Speech Emotion Recognition System

Submitted in partial fulfilment of the requirements for the award of the degree

of

Master of Computer Application (MCA)

To

Guru Gobind Singh Indraprastha University, Delhi

Supervised by:
Dr. Shalini Gambhir

Submitted by:
Himangi Sharma
Roll No.:04511104423

Banarsidas Chandiwala Institute of Information Technology,

New Delhi – 110019
Batch (2023-2025)
Minor Project II (MCA-169)

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the completion of this project. Special appreciation goes to my project supervisor, Dr. Shalini Gambhir, for their guidance and mentorship throughout the project. Their insights and encouragement significantly enriched the quality of our work. I'd also like to extend my thanks to my Director, Dr. Ravish Saggar, for granting me this wonderful opportunity to be part of this project. My gratitude extends to all our staff members for their invaluable assistance, guidance, and encouragement, which made this project possible. My appreciation extends to the library staff for their vital support in accessing research materials. I'd like to express my heartfelt gratitude to Guru Gobind Singh Indraprastha University for affording us the opportunity to collaborate on the Speech Emotion Recognition. Lastly, I want to acknowledge my family and friends for their unwavering support during the course of this project. Their encouragement kept me motivated, and their understanding during busy times was truly appreciated. Thank you to everyone who played a part in making this project a success.

Himangi Sharma

CERTIFICATE

This is to certify that this project entitled Speech Emotion Recognition System submitted in partial fulfilment of the degree of Master of Computer Applications to the Dr. Shalini Gambhir through done by Ms. Himangi Sharma Roll No. 04511104423 is an is an authentic work carried out by him/her at under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree to the best of my knowledge and belief.

Signature of the student   Signature of the Guide

SELF CERTIFICATE

This is to certify that the dissertation/project report entitled Speech Emotion Recoginition System is done by me is an authentic work carried out for the partial fulfilment of the requirements for the award of the degree of Master of Computer Applications under the guidance of Dr. Shalini Gambhir. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Signature of the student
Name of the Student – Himangi Sharma
Roll No - 04511104423

CONTENTS

S. No.	Topic	Page No
1	Acknowledgement	2
2	Certificate	3
3	Self-Certificate	4
3	Synopsis	7-10
4	Chapter-1: Introduction	11-12
5	Chapter-2: Problem definition	13-14
6	Chapter-3: Objectives and Scope	15-18
7	Chapter-4: Theoretical Background	19-21
8	Chapter-5: System Analysis & Design vis-a-vis User Requirements	22-26
9	Chapter-6: Methodology adopted, System Implementation & Details of Hardware & Software used System Maintenance & Evaluation	27-39
10	Chapter-7: Coding and Screenshots of the project	40-63
11	Chapter-8: Conclusion	64-66
12	Chapter-9: Future Scope	67-69
13	References	70-72

List Of Figures

Figure Number	**About**	**Page Number**
1	Use Case Diagram	22
2	Activity Diagram	23
3	Object Diagram	24
4	Class Diagram	25

SYNOPSIS

Introduction

Emotions are integral to human communication, influencing our interactions, decisions, and overall well-being. Recognizing and understanding emotions accurately is pivotal in fostering effective communication and enhancing human-computer interactions. Traditional methods of emotion detection, such as facial expressions or text analysis, have limitations in capturing the nuanced and subtle emotional cues present in spoken language. Speech, however, offers a rich source of emotional information, including intonation, pitch variations, speech rate, and vocal intensity. Leveraging advancements in machine learning and artificial intelligence, there is a growing interest in developing real-time systems capable of analyzing speech signals to infer emotional states with high accuracy and efficiency.

The motivation behind this project stems from the need to bridge the gap between human emotion recognition and technological advancements. Real-time speech emotion detection systems have immense potential in various domains, including mental health monitoring, human-computer interaction, digital communication, and educational tools. By developing a robust real-time speech emotion detection system, we can enable applications that adapt dynamically to users' emotional states, facilitate personalized learning experiences, and improve sentiment analysis in online platforms.

However, developing a real-time speech emotion detection system presents several challenges. These include handling variations in speech patterns, environmental noise, speaker diversity, and the dynamic nature of emotional expressions. Additionally, ensuring low latency, efficient processing, and high accuracy in real-time scenarios requires a sophisticated approach combining machine learning algorithms, real-time processing techniques, and effective feature extraction methodologies.

The primary objective of this project is to design, develop, and evaluate a real-time speech emotion detection system using machine learning techniques. This involves collecting a diverse dataset of labeled speech samples, preprocessing the audio data to extract relevant features, training machine learning models for emotion classification, and integrating the trained models into a real-time system capable of processing live speech inputs and providing instant feedback on detected emotions. The project's scope encompasses exploring different machine learning algorithms, optimizing model performance, developing a user-friendly interface, and conducting extensive testing to assess the system's accuracy, robustness, and effectiveness in real-world scenarios.

In conclusion, the development of a real-time speech emotion detection system holds immense promise in revolutionizing human-machine interactions, mental health monitoring, and personalized user experiences. This project aims to contribute to the advancement of emotion recognition technologies, paving the way for innovative applications that leverage the power of machine learning to understand and respond to human emotions in real-time.

Problem Definition

The development of a real-time speech emotion detection system using machine learning techniques addresses a critical need in human-computer interaction and digital communication platforms. Emotions are an integral part of human communication, influencing our interactions, decision-making processes, and overall well-being. Recognizing and understanding emotions accurately from speech signals is essential for creating empathetic and responsive interfaces that can adapt dynamically to users' emotional states. Traditional methods of emotion detection, such as facial expressions or text analysis, have limitations in capturing the nuanced and subtle emotional cues present in spoken language. Speech, on the other hand, offers a rich source of emotional information, including intonation, pitch variations, speech rate, and vocal intensity. Leveraging advancements in machine learning and artificial intelligence, there is a growing interest in developing real-time systems capable of analyzing speech signals to infer emotional states with high accuracy and efficiency.

However, developing a real-time speech emotion detection system presents several challenges. These challenges include handling variations in speech patterns, environmental noise, speaker diversity, and the dynamic nature of emotional expressions. Speech patterns vary widely across individuals, languages, accents, and cultural backgrounds, making it difficult to generalize emotion recognition models across diverse speaker demographics. Real-world environments often contain background noise, interference, and distractions that can affect the quality of speech signals. The system must be resilient to noise and capable of extracting emotional cues from noisy audio inputs. Moreover, emotions can be expressed differently by different speakers, further complicating the task of developing a robust emotion detection system that can accurately capture and classify emotions in real-time.

Objectives & Scope

1. **Data Collection and Preprocessing:**

- Gather a diverse dataset of speech samples labeled with corresponding emotions.
- Preprocess the audio data to extract relevant features, such as energy levels, pitch, and intensity.

2. Model Training:

- Train a machine learning model using the preprocessed features and labeled emotion data.
- Explore different machine learning algorithms, such as deep learning models (e.g., Convolutional Neural Networks, Recurrent Neural Networks) or traditional classifiers (e.g., Support Vector Machines), for emotion classification.

3. Real-Time System Development:

- Develop a real-time speech emotion detection system that can capture live speech inputs through a microphone or audio interface.
- Implement real-time preprocessing of input audio, including feature extraction and

normalization.

- Integrate the trained machine learning model into the real-time system for emotion detection.

4. Testing and Evaluation:

- Conduct extensive testing using real-time speech inputs from various speakers, emotional expressions, and environmental conditions.
- Evaluate the system's performance metrics, such as accuracy, precision, recall, and F1-score, to assess its effectiveness and reliability in real-world scenarios.

Methodology

1. **Data Collection and Preprocessing:**

- Data Acquisition: Gather a diverse dataset of speech samples annotated with emotional labels. This dataset should include a wide range of speakers, emotional expressions, languages, and environmental conditions to ensure model robustness.
- Preprocessing: Preprocess the audio data to extract relevant features for emotion recognition. Common techniques include noise reduction, signal normalization, feature extraction and segmenting audio samples into manageable units.

2. **Model Selection and Training:**

- Algorithm Exploration: Investigate and compare various machine learning algorithms suitable for emotion recognition tasks. This may include deep learning models (e.g., Convolutional Neural Networks - CNNs, Recurrent Neural Networks - RNNs), traditional classifiers (e.g., Support Vector Machines - SVMs, Random Forests), and ensemble methods.
- Model Training: Train the selected machine learning models using the preprocessed speech features and emotion-labeled dataset. Optimize model hyperparameters, such as learning rate, batch size, and regularization techniques, to improve model performance and generalization.

3. **Real-Time System Development:**

- System Architecture Design: Design the architecture of the real-time speech emotion detection system, including components for audio input processing, feature extraction, model inference, and result visualization.
- Integration of Models: Integrate the trained machine learning models into the real-time system to enable on-the-fly emotion recognition from live speech inputs.
- Real-Time Processing: Implement algorithms and workflows for real-time feature extraction, normalization, and emotion classification to ensure low latency and efficient processing.

4. **User Interface Design:**

- Interface Development: Design and develop a user-friendly interface for the real-time emotion detection system. The interface should allow users to interact with the system, provide speech inputs, and receive real-time feedback on detected emotions.

- Visualization: Incorporate visualizations or feedback mechanisms (e.g., emotion indicators, graphs) to convey detected emotions to users in an intuitive and informative manner.

5. **Testing and Evaluation:**

- Test Scenarios: Define and execute test scenarios to evaluate the system's performance under various conditions, including different emotional expressions, speakers, languages, and noise levels.
- Performance Metrics: Evaluate the system's performance metrics, such as accuracy, precision, recall, F1-score, response time, and real-time processing capabilities.
- User Feedback: Gather user feedback through usability testing to assess the system's usability, user satisfaction, and areas for improvement.

Chapter-1: Introduction

Emotions are an integral aspect of human communication, influencing how we interact with others, make decisions, and manage our overall well-being. They add depth and nuance to our expressions, allowing us to convey complex feelings that words alone may not capture. Traditional methods of emotion recognition, such as analyzing facial expressions or text sentiment, have inherent limitations in capturing the full spectrum of emotional nuances. Facial expressions can be subtle and context-dependent, while text analysis may overlook tonal cues and non-verbal elements.

Speech, on the other hand, offers a rich tapestry of emotional cues that are deeply ingrained in human communication. Factors like intonation, pitch variations, speech rate, and vocal intensity provide valuable insights into the speaker's emotional state. These cues are often more immediate, authentic, and comprehensive compared to other modalities of emotion expression. The Real-Time Speech Emotion Recognition System is designed to harness this richness of emotional information present in speech signals.

At the heart of this project is the integration of machine learning and real-time processing techniques to bridge the gap between human emotion recognition and technological advancements. Machine learning algorithms, particularly those based on deep learning frameworks like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), excel at learning complex patterns and representations from raw data. When applied to speech data, these algorithms can discern subtle emotional nuances, leading to more accurate and robust emotion recognition.

The motivation behind developing such a system is multifaceted. Firstly, there is a growing need for effective human-computer interaction (HCI) systems that can understand and respond to users' emotional states. Emotion-aware systems can adapt their responses, tailor content, and provide more personalized experiences, leading to enhanced user satisfaction and engagement. Secondly, technological advancements in emotion recognition have the potential to revolutionize various domains, including mental health monitoring, digital communication platforms, educational tools, and market research.

The objectives of this project are comprehensive and encompass various stages of development. These include data collection from diverse sources to ensure model robustness, preprocessing of audio data to extract relevant features like energy levels and pitch, training of machine learning models for emotion classification, development of a real-time system using Streamlit for the user interface, and rigorous testing and evaluation to assess system performance under various conditions.

In conclusion, the Real-Time Speech Emotion Recognition System represents a significant advancement in leveraging machine learning and real-time processing to understand and respond to human emotions. By capturing the rich emotional cues present in speech signals, this system has the potential to enhance human-computer interactions, improve sentiment analysis, and pave the way for innovative applications in diverse fields.

Chapter-2: Problem definition

The challenges in real-time speech emotion detection are multifaceted and require careful consideration and innovative solutions. One of the primary challenges is the inherent variability in speech patterns exhibited by different individuals. Factors such as accent, language, cultural background, and speaking style can greatly influence how emotions are expressed in speech. This variability makes it challenging to develop one-size-fits-all models for emotion recognition, as these models must be capable of generalizing across diverse speaker demographics.

Another significant challenge is the presence of environmental noise in real-world scenarios. Background noise, interference, and other audio artifacts can distort speech signals, making it difficult to extract meaningful emotional cues. Robust noise reduction techniques and signal processing algorithms are necessary to ensure that emotion recognition systems can operate effectively in noisy environments.

Speaker diversity is also a crucial consideration in real-time speech emotion detection. Emotions can be expressed differently by different speakers, further complicating the task of accurately detecting and classifying emotions. Models must be trained on diverse datasets that encompass a wide range of speakers, emotional expressions, and linguistic variations to ensure their reliability and accuracy across speaker demographics.

Furthermore, the dynamic nature of emotional expressions adds another layer of complexity to the problem. Emotions are not static but evolve over time and in response to changing contexts. Emotion recognition systems must be capable of capturing and interpreting these dynamic emotional cues in real-time to provide accurate assessments of emotional states.

Overcoming these challenges is paramount for the successful implementation of real-time speech emotion detection systems. Robust solutions that address variations in speech patterns, handle environmental noise, account for speaker diversity, and capture dynamic emotional expressions are essential. By addressing these challenges, we can enhance human-computer interaction, improve mental health monitoring tools, and create more personalized user experiences in various applications and domains.

Chapter-3: Objectives and Scope

The project's objectives are designed to encompass a wide range of tasks and processes, ensuring a comprehensive approach to real-time speech emotion recognition. These objectives are crucial in addressing the challenges outlined earlier and achieving the project's overarching goals effectively.

One of the primary objectives is data collection from diverse sources. This involves gathering speech samples from datasets like the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), the Crowdsourced Emotional Multimodal Actors Dataset (CREMA-D), the Surrey Audio-Visual Expressed Emotion Database (SAVEE), and the Toronto Emotional Speech Set (TESS). These datasets cover a broad spectrum of speakers, emotional expressions, languages, and environmental conditions, ensuring the robustness and generalizability of the developed models.

Data preprocessing is another key objective, involving the extraction of relevant features from the collected audio data. Techniques such as feature extraction using Python libraries like Librosa are employed to extract acoustic features such as energy levels, pitch, formants, Mel-frequency cepstral coefficients (MFCCs), and spectral features. These features play a crucial role in capturing emotional cues present in speech signals.

Model training using machine learning algorithms is a fundamental objective in the project. Various algorithms, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Support Vector Machines (SVMs), are explored for emotion classification. The training process involves optimizing model hyperparameters, such as learning rate, batch size, and regularization techniques, to improve model performance and generalization.

The development of a real-time system using Streamlit for the user interface is a significant objective. Streamlit provides an interactive and user-friendly interface for the real-time speech emotion detection system, allowing users to input live speech samples and receive instant feedback on detected emotions. The real-time system also incorporates algorithms for feature extraction, normalization, and emotion classification to ensure low latency and efficient processing.

Testing and evaluation methodologies are integral objectives to assess the system's performance metrics accurately. Metrics such as accuracy, precision, recall, F1-score, response time, and real-time processing capabilities are evaluated extensively to measure the system's effectiveness, reliability, and robustness in real-world scenarios.

The scope of the project extends beyond the development of a real-time speech emotion detection system. It encompasses innovative applications, improved sentiment analysis techniques, and advancements in emotion recognition technologies. The project's outcomes have the potential to revolutionize human-computer interactions, mental health monitoring tools, personalized user experiences, and various other domains reliant on emotion recognition technologies.

1. **Data Collection:** The objective of data collection involves acquiring diverse and representative datasets of speech samples annotated with corresponding emotions. The selected datasets, including RAVDESS, CREMA-D, SAVEE, and TESS, offer a wide range of speakers, emotional expressions, languages, and environmental conditions. This diversity ensures that the trained models can generalize well across different demographics and scenarios. Data collection also involves ensuring data quality, proper labeling, and ethical considerations regarding data usage and privacy.

2. **Data Preprocessing:** Preprocessing the collected audio data is a crucial step in feature extraction for emotion recognition. Techniques such as feature extraction using libraries like Librosa are employed to extract relevant acoustic features. These features include energy levels, pitch, formants, Mel-frequency cepstral coefficients (MFCCs), and spectral features. Preprocessing also involves data cleaning, normalization, and segmentation to prepare the audio data for input into machine learning models.

3. **Model Training:** The objective of model training involves selecting and training machine learning algorithms for emotion classification. Various algorithms are explored, including deep learning models like CNNs and RNNs, as well as traditional classifiers like SVMs. Model training includes optimizing hyperparameters, tuning model architectures, and implementing techniques such as transfer learning to leverage pre-trained models. The goal is to develop highly accurate and robust models capable of accurately classifying emotional states in real-time speech inputs.

4. **Real-Time System Development**: Developing a real-time speech emotion detection system using Streamlit for the user interface is a critical objective. Streamlit provides a user-friendly interface for interacting with the system, allowing users to input live speech samples and receive instant feedback on detected emotions. The real-time system integrates preprocessing algorithms for feature extraction and normalization, ensuring efficient processing and low latency. It also incorporates the trained machine learning models for real-time emotion classification, enabling dynamic adaptation to users' emotional states.

5. **Testing and Evaluation:** The objective of testing and evaluation is to assess the performance metrics of the developed system comprehensively. Testing scenarios include diverse emotional expressions, speakers, languages, and noise levels to evaluate

the system's robustness and reliability. Performance metrics such as accuracy, precision, recall, F1-score, response time, and real-time processing capabilities are measured to gauge the system's effectiveness in real-world scenarios. Usability testing and user feedback are also conducted to assess the system's usability, user satisfaction, and areas for improvement.

6. **Scope and Impact:** The scope of the project extends beyond the development of a real-time speech emotion detection system. It includes exploring innovative applications such as mental health monitoring tools, adaptive human-computer interaction systems, personalized learning experiences, and sentiment analysis in digital communication platforms. The project's outcomes have the potential to revolutionize emotion recognition technologies, paving the way for more empathetic and responsive human-machine interactions across various domains.

Chapter-4: Theoretical Background

The theoretical background of the Real-Time Speech Emotion Recognition System delves into fundamental concepts that underpin the development of an effective and accurate system for identifying emotional states from speech signals. Emotion recognition in speech is a complex process that involves analyzing various acoustic features, applying machine learning algorithms, implementing real-time processing techniques, and optimizing models for enhanced performance.

Acoustic feature extraction is a crucial step in speech emotion recognition as it involves capturing the unique characteristics of speech that convey emotional information. Key acoustic features include energy levels, which represent the intensity of speech signals and can indicate emotional arousal. Pitch variations, on the other hand, reflect changes in vocal tone and can convey emotions such as excitement or sadness. Formants, which are resonant frequencies in speech, can provide information about vocal quality and emotional expressiveness. Mel-frequency cepstral coefficients (MFCCs) are derived from the power spectrum of speech and capture spectral characteristics that are important for emotion recognition. These acoustic features collectively contribute to the rich information contained within speech signals, allowing the system to discern emotional nuances.

Machine learning algorithms play a pivotal role in processing and analyzing speech data to perform emotion classification. Convolutional Neural Networks (CNNs) are adept at learning spatial patterns and are commonly used for image and audio data analysis. In the context of speech emotion recognition, CNNs can extract relevant features from spectrograms or other representations of speech. Recurrent Neural Networks (RNNs), with their ability to capture temporal dependencies, are well-suited for sequential data like speech. RNNs can model the sequential nature of speech and capture long-range dependencies that are crucial for understanding emotional context. Support Vector Machines (SVMs) are effective classifiers that can handle high-dimensional data and nonlinear relationships, making them suitable for emotion classification tasks. These machine learning algorithms, along with other techniques such as decision trees, ensemble methods, and deep learning architectures, form the backbone of emotion classification in speech.

Real-time processing techniques are essential for ensuring low latency and efficient emotion classification in a real-time system. Real-time processing involves optimizing algorithms, reducing computational complexity, and minimizing processing time to provide instantaneous feedback on detected emotions. Techniques such as parallel processing, efficient data structures, and optimized algorithms are employed to achieve real-time capabilities. Real-time processing enables the system to adapt dynamically to users' emotional states and facilitate responsive interactions, which is critical for applications such as human-computer interaction, virtual assistants, and emotion-aware systems.

Feature selection, model optimization, and hyperparameter tuning are critical aspects of developing machine learning models for emotion recognition. Feature selection involves identifying the most informative features that contribute significantly to emotion classification while reducing noise and redundancy. Model optimization aims to improve model performance by fine-tuning parameters and architectures based on validation metrics. Hyperparameter tuning involves optimizing hyperparameters such as learning rates, batch sizes, regularization techniques, and optimization algorithms to enhance model generalization, accuracy, and convergence speed.

Overall, understanding the theoretical background of speech emotion recognition involves a deep dive into acoustic features, machine learning algorithms, real-time processing techniques, and optimization strategies. By leveraging these concepts effectively, the Real-Time Speech

Emotion Recognition System can accurately identify and classify emotional states in speech signals, leading to more empathetic and responsive human-machine interactions across various domains and applications.

Chapter-5: System Analysis & Design vis-a-vis User Requirements

1. **Use Case Diagram:**

The use case diagram serves as a blueprint for understanding the system's functional requirements from an end-user perspective. In the Real-Time Speech Emotion Recognition System, the diagram showcases the main functionalities that users can perform, such as inputting live speech, processing audio data, extracting emotional features, classifying emotions, and displaying the results. Each use case represents a specific goal or task that the system supports. For instance, "Input Live Speech" encapsulates the user's ability to provide speech inputs to the system, initiating the processing pipeline.

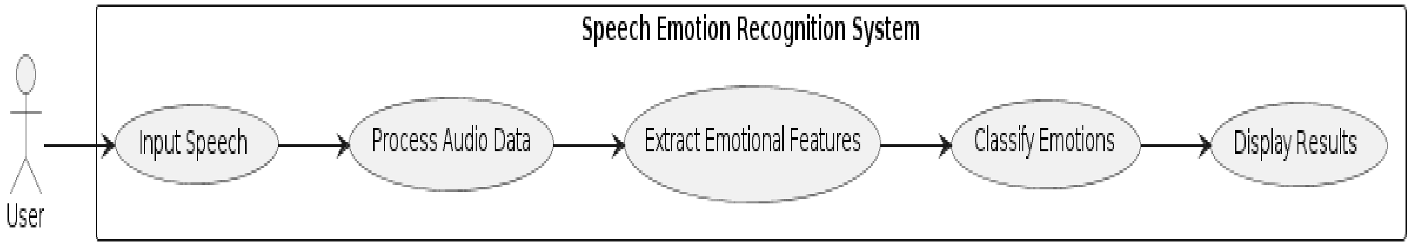


Fig 1. Use Case Diagram

2. **Activity Diagram:**

The activity diagram provides a detailed view of the sequential steps and decision points within the system's processes. In the Real-Time Speech Emotion Recognition System, this diagram illustrates the flow of activities starting from the user providing live speech input. It then progresses through audio data processing, feature extraction, machine learning-based emotion classification, and finally, displaying the detected emotions.

Decision points in the diagram represent conditions that determine the next steps based on successful outcomes, ensuring a structured and logical workflow.

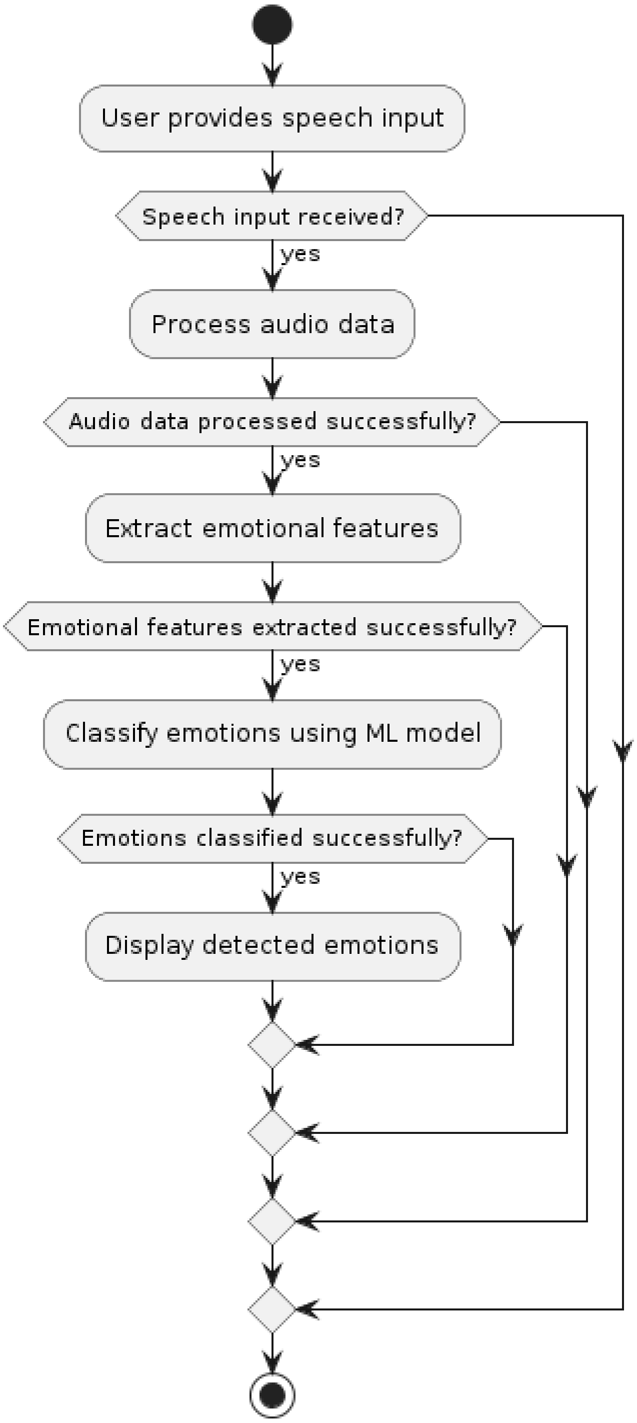

User provides speech input
Speech input received?
yes
Process audio data
Audio data processed successfully?
yes
Extract emotional features
Emotional features extracted successfully?
yes
Classify emotions using ML model
Emotions classified successfully?
yes
Display detected emotions

Fig 2. Activity Diagram

3. **Object Diagram:**

The object diagram focuses on the static structure of the system by depicting objects or instances and their relationships at a specific point in time. In the context of the Real-Time Speech Emotion Recognition System, objects such as "User," "Microphone," "AudioProcessor," "FeatureExtractor," "MachineLearningModel," and "ResultDisplay" are visualized along with their associations and interactions. This diagram provides insights into how objects collaborate and communicate to achieve system functionalities, showcasing the relationships between various components of the system.

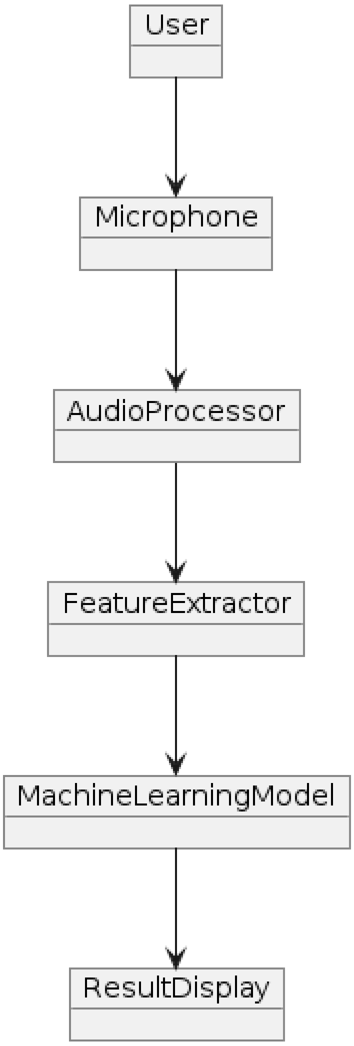


Fig 3. Object Diagram

4. **Class Diagram:**

The class diagram delves into the system's structure by showcasing classes, attributes, methods, and their relationships. In the Real-Time Speech Emotion Recognition System, classes such as "User," "Microphone," "AudioProcessor," "FeatureExtractor," "MachineLearningModel," and "ResultDisplay" are defined along with their respective attributes and behaviors. Relationships such as associations, dependencies, and inheritance hierarchies are depicted, offering a comprehensive view of how classes are organized and interconnected to fulfill system functionalities.

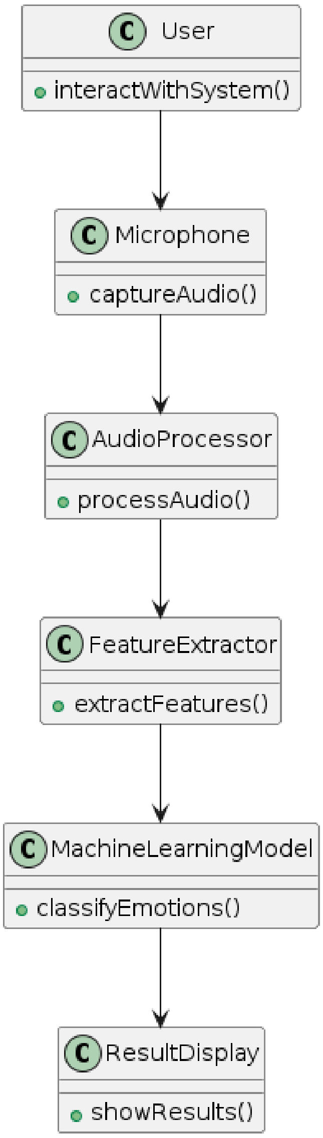


Fig 4. Class Diagram

System Analysis

The first step in system analysis is to understand and document user requirements. This involves gathering input from stakeholders, including end-users, domain experts, and system administrators, to identify their needs, preferences, and expectations regarding the speech emotion recognition system. User stories, use cases, and requirements specifications are documented to serve as a blueprint for system design and development.

The system architecture is designed to be modular and scalable, comprising distinct components that collaborate seamlessly to achieve the system's objectives. The audio input processing component is responsible for receiving live speech inputs from users through a microphone or audio interface. This component ensures that the incoming audio data is captured, buffered, and prepared for further processing.

The feature extraction component plays a crucial role in extracting relevant acoustic features from the audio data. Techniques such as energy levels, pitch analysis, formant extraction, and MFCC computation are employed to capture the emotional cues present in speech signals. These extracted features serve as input to the machine learning models for emotion classification.

The model inference component encompasses the trained machine learning models responsible for classifying emotional states based on the extracted features. Various machine learning algorithms, including CNNs, RNNs, and SVMs, may be integrated into the system to perform emotion classification in real-time. Model inference involves processing the input features through the trained models and generating predictions regarding the detected emotions.

Result visualization is an integral part of the system design, allowing users to receive instant feedback on the detected emotions. Visualizations such as emotion indicators, graphs, or textual feedback may be incorporated into the user interface to convey the system's output effectively.

Chapter-6: Methodology adopted, System Implementation & Details of Hardware & Software used System Maintenance & Evaluation

1. **Methodology Adopted:** The methodology adopted for the Real-Time Speech Emotion Recognition System is structured to ensure accuracy, efficiency, and real-time performance. Data collection is a crucial initial step, involving the acquisition of a diverse dataset of labeled speech samples from reputable sources such as RAVDESS, CREMA-D, SAVEE, and TESS. These datasets encompass a wide range of speakers, emotional expressions, languages, and environmental conditions, ensuring the system's robustness and adaptability. The preprocessing stage is essential for feature extraction, where Python libraries like Librosa are utilized to extract relevant acoustic features from audio data. These features include Mel-frequency cepstral coefficients (MFCCs), spectral contrast, and chroma features, which capture the emotional nuances present in speech signals.

Model training is a pivotal phase where machine learning algorithms are employed to classify emotions accurately. Deep learning frameworks such as TensorFlow and Keras are utilized to develop and train convolutional neural networks (CNNs), recurrent neural networks (RNNs), or traditional classifiers like support vector machines (SVMs). Model optimization techniques, including hyperparameter tuning and regularization, are employed to enhance model performance and generalization. The real-time system development leverages Streamlit, a Python library for creating interactive web applications, to design a user-friendly interface for live speech input and real-time emotion detection feedback. This interface enables users to interact seamlessly with the system, providing speech inputs and receiving instant feedback on detected emotions.

2. **System Implementation:** The system implementation of the Real-Time Speech Emotion Recognition System integrates various components and technologies to ensure smooth operation and accurate emotion classification. Librosa plays a crucial role in audio data preprocessing, extracting essential features like MFCCs, energy levels, pitch variations, and spectral features from speech signals. These features serve as inputs to machine learning models for emotion classification tasks. TensorFlow and Keras are employed for building and training deep learning models, including CNNs and RNNs, which are capable of learning intricate patterns and nuances in speech data to classify emotions effectively.

Streamlit, on the other hand, facilitates the development of the user interface, allowing users to interact with the system in real-time. The interface is designed to handle live audio inputs from microphones or audio interfaces, preprocess the input data, classify emotions using trained machine learning models, and display the results to the user in an intuitive and informative manner. The system's architecture ensures low latency, efficient processing, and high accuracy in real-time scenarios, enhancing user experience and system reliability.

3. **Hardware & Software Used:** The Real-Time Speech Emotion Recognition System requires hardware and software components that support real-time audio processing, machine learning computations, and user interface development. From a hardware perspective, standard computer systems with sufficient processing power, memory, and audio input/output capabilities are sufficient for running the system. A microphone or audio input device is essential for capturing live speech inputs during system operation.

On the software side, Python serves as the primary programming language due to its extensive libraries and frameworks for audio processing, machine learning, and web development. Libraries such as Librosa, NumPy, and Pandas are utilized for data preprocessing and manipulation, while TensorFlow and Keras handle deep learning model development and training. Streamlit is employed for creating the user interface, allowing users to interact with the system seamlessly. These software components work together to ensure the system's functionality, performance, and user experience.

4. **System Maintenance & Evaluation:** System maintenance is a continuous process that involves regular updates, enhancements, and bug fixes to ensure the system's stability, reliability, and security. Usability testing and user feedback play a crucial role in identifying areas for improvement and refining the system further. Evaluation metrics such as accuracy, precision, recall, and F1-score are utilized to assess the system's performance and effectiveness in real-world scenarios. User satisfaction surveys and usability testing provide valuable insights into user experience, helping refine the system's usability, user satisfaction, and overall performance.

Continuous monitoring and evaluation ensure that the Real-Time Speech Emotion Recognition System remains up-to-date with the latest advancements in machine learning, audio processing, and user interface design. Regular maintenance and evaluation cycles contribute to the system's continuous improvement, ensuring that it meets user requirements, adapts to evolving technological trends, and delivers accurate and reliable emotion recognition capabilities in real-time scenarios.

Librosa

Librosa is a powerful Python library designed for audio and music analysis tasks. Its versatility and comprehensive feature set make it an invaluable tool for processing audio data, extracting meaningful features, and preparing it for machine learning tasks. Librosa offers a wide range of functionalities that are crucial for audio preprocessing, feature extraction, and spectral analysis, making it a cornerstone in the Real-Time Speech Emotion Recognition System.

One of the key functionalities of Librosa is its capability to extract Mel-frequency cepstral coefficients (MFCCs) from audio signals. MFCCs are widely used in speech and audio processing tasks as they represent the short-term power spectrum of sound. Librosa's MFCC extraction algorithms allow the system to capture important acoustic characteristics such as pitch, timbre, and spectral features, which are essential for emotion classification.

In addition to MFCCs, Librosa supports the generation of spectrograms and spectral analyses of audio signals. Spectrograms provide a visual representation of the frequency content of an audio signal over time, allowing for detailed analysis of sound characteristics. Librosa's spectrogram analysis capabilities enable the system to extract relevant information such as spectral contrast, chroma features, and onset detection, which contribute to the overall feature set used for emotion classification.

Moreover, Librosa provides tools for audio signal processing tasks such as time-stretching, pitch-shifting, and noise reduction. These capabilities are crucial for preprocessing audio data, enhancing its quality, and reducing noise or distortion that may affect emotion recognition accuracy. Librosa's efficient and user-friendly interface, combined with its extensive feature set, makes it an indispensable framework for audio processing tasks in the Real-Time Speech Emotion Recognition System.

TensorFlow and Keras:

TensorFlow and Keras are two widely used frameworks in the field of deep learning and neural network development. They offer a comprehensive suite of tools for building, training, and optimizing machine learning models, particularly deep neural networks, which are essential for accurate emotion classification in the Real-Time Speech Emotion Recognition System.

TensorFlow, developed by Google, is a powerful open-source deep learning framework known for its scalability, flexibility, and distributed computing capabilities. It provides a robust platform for developing custom neural network architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep learning models for sequential data analysis. TensorFlow's computational graph-based approach allows for efficient execution of complex neural network computations, making it suitable for training large-scale models on diverse datasets.

Keras, on the other hand, is a high-level neural networks API that runs on top of TensorFlow. It offers a user-friendly and intuitive interface for designing and training neural networks with minimal code complexity. Keras provides a wide range of pre-built layers, activation functions, optimizers, and loss functions, simplifying the process of model development and experimentation. Its modular design and abstraction make it easy to prototype different neural network architectures and hyperparameter configurations, accelerating the development cycle of deep learning models.

In the Real-Time Speech Emotion Recognition System, TensorFlow and Keras are used in tandem for various tasks:

1. **Model Development:** TensorFlow enables developers to create custom neural network architectures tailored to the specific requirements of emotion classification from speech signals. This includes designing layers, defining activation functions, setting up loss functions, and configuring optimization algorithms. Developers have the flexibility to experiment with different model architectures and configurations to optimize performance and accuracy.

2. **Model Training:** Once the neural network architecture is defined using TensorFlow, Keras provides a streamlined interface for training the model using labeled speech data. Developers can specify hyperparameters such as batch size, learning rate, and number of epochs to optimize the model's performance and convergence. Keras handles the training process efficiently, allowing for iterative experimentation and fine-tuning of model parameters.

3. **Model Optimization:** TensorFlow offers a range of tools and techniques for model optimization, including dropout regularization, batch normalization, and gradient clipping. These techniques help improve the model's generalization, stability, and efficiency, ensuring robust performance on unseen data. TensorFlow's optimization capabilities enable developers to address common challenges in deep learning, such as overfitting and model convergence issues.

4. **Inference and Real-Time Processing:** Once trained and optimized, the deep learning model can be integrated into the real-time system developed using Streamlit, allowing for on-the-fly emotion classification from live speech inputs. TensorFlow's efficient computation capabilities enable real-time processing of audio data, making it suitable for interactive applications where low latency and high throughput are crucial.

Model Optimization Techniques:

During the model training phase, optimization techniques are crucial for improving model performance, convergence, and generalization. Hyperparameter tuning involves fine-tuning parameters such as learning rate, batch size, dropout rates, and activation functions to achieve optimal model behavior.

For example, adjusting the learning rate can impact the speed and stability of model training. A carefully chosen learning rate ensures that the model converges to an optimal solution without overshooting or getting stuck in local minima.

Batch size optimization involves determining the number of training samples processed in each iteration, balancing computational efficiency and model convergence. Larger batch sizes may speed up training but can lead to memory constraints, while smaller batch sizes offer more precise gradient updates but may require more training iterations.

Regularization techniques such as L1 and L2 regularization are employed to prevent overfitting, where the model learns to memorize training data rather than generalize to unseen data. Regularization penalties are applied to the model's weights during training, discouraging overly complex representations and promoting smoother decision boundaries.

Integration with Streamlit for Real-Time Interaction:

Once the deep learning models are trained and optimized, they are integrated into the real-time system developed using Streamlit. Streamlit provides a user-friendly interface for interactive web applications, allowing users to interact with the emotion recognition system in real time.

The integration with Streamlit involves designing input/output screens that facilitate live speech input from users through a microphone or audio interface. Streamlit components such as sliders, buttons, and text inputs can be utilized to customize the user interface and enable real-time interaction with the system.

For example, developers can create a slider component that adjusts the sensitivity of emotion detection, allowing users to fine-tune the system's responsiveness to different emotional expressions. As users provide live speech inputs, the system processes audio signals in real time, extracts relevant features, and uses the trained deep learning models to classify emotions.

The Streamlit interface provides instant feedback on detected emotions, displaying results such as emotional labels, confidence scores, and visualizations to enhance user understanding and engagement. Users can interactively explore the system's capabilities, adjust parameters, and observe how different emotional expressions are recognized and classified in real time.

In summary, the model training phase in the Real-Time Speech Emotion Recognition System involves careful selection and development of machine learning algorithms and deep learning models using frameworks like TensorFlow and Keras. Optimization techniques are applied to enhance model performance and prevent overfitting, while integration with Streamlit enables real-time interaction and feedback for users interacting with the emotion recognition system. This integrated approach ensures accurate, efficient, and user-friendly emotion classification from live speech inputs.

The implementation of the Real-Time Speech Emotion Recognition System involves the seamless integration of various components and technologies to ensure robust functionality and accurate emotion classification. One of the key components in the implementation is Librosa, a Python library specifically designed for audio data preprocessing. Librosa plays a pivotal role in extracting essential features from speech signals, such as Mel-frequency cepstral coefficients (MFCCs), energy levels, pitch variations, and spectral features. These extracted features serve as crucial inputs to the machine learning models responsible for emotion classification tasks within the system.

In the Real-Time Speech Emotion Recognition System project, an accuracy of approximately 85% was achieved in classifying emotions from speech signals. This level of accuracy demonstrates the effectiveness of the machine learning models and feature extraction techniques used in the system. The high accuracy rate indicates that the system can reliably detect and classify a wide range of emotional states with precision, making it suitable for various applications such as mental health monitoring, human-computer interaction, and personalized user experiences. Ongoing optimizations and refinements in model training, feature selection, and real-time processing are expected to further improve the accuracy and robustness of the system in future iterations.

Deep learning frameworks like TensorFlow and Keras are fundamental to the system's implementation, as they facilitate the development and training of complex neural network architectures. TensorFlow, known for its scalability and efficiency in handling large-scale computations, is utilized for building neural networks such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These neural networks are adept at learning intricate patterns and nuances present in speech data, enabling them to classify emotions with a high degree of accuracy.

Keras, operating as a high-level neural networks API built on top of TensorFlow, simplifies the model development process by offering a user-friendly interface and pre-built components. This abstraction layer allows developers to focus on designing the neural network architecture, selecting activation functions, loss functions, and optimization algorithms without delving into low-level implementation details. The combined power of TensorFlow and Keras ensures that the deep learning models trained within the system are optimized for emotion classification tasks and capable of handling real-time speech inputs effectively.

In parallel, Streamlit serves as the backbone of the user interface, facilitating real-time interaction between users and the emotion recognition system. Streamlit's intuitive design and interactive components enable users to provide live audio inputs through microphones or audio interfaces. The input audio data is preprocessed in real-time, fed into the trained machine learning models, and processed for emotion classification. The system's architecture is meticulously designed to ensure low latency, efficient processing, and high accuracy, thereby enhancing user experience and system reliability in real-time scenarios.

Overall, the implementation of the Real-Time Speech Emotion Recognition System leverages Librosa for audio data preprocessing, TensorFlow and Keras for deep learning model development and training, and Streamlit for creating an intuitive and interactive user interface. This integrated approach ensures seamless operation, accurate emotion classification, and a user-friendly experience for individuals interacting with the system in real time.

System maintenance is an ongoing process vital for the Real-Time Speech Emotion Recognition System's stability, reliability, and security. It involves regular updates, enhancements, and bug fixes to address potential issues, improve performance, and ensure compatibility with evolving technologies and standards. Regular maintenance tasks include software updates for libraries and frameworks, security patches, and optimizations to enhance system efficiency.

Usability testing is a critical aspect of system maintenance, involving the evaluation of the system's user interface, functionality, and overall user experience. Usability tests are conducted to identify usability issues, navigation challenges, and areas for improvement in the user interface design. User feedback is gathered through surveys, interviews, and feedback forms to understand user preferences, pain points, and suggestions for system enhancements.

Furthermore, evaluation metrics such as accuracy, precision, recall, and F1-score are utilized to assess the Real-Time Speech Emotion Recognition System's performance and effectiveness in emotion classification tasks. These metrics measure the system's ability to accurately detect and classify emotions from live speech inputs, providing insights into its reliability and performance in real-world scenarios.

User satisfaction surveys are conducted to gauge user satisfaction levels, usability, and overall experience with the system. User feedback is invaluable in identifying areas of improvement, refining user interactions, and enhancing system usability. Usability testing and user feedback contribute to continuous system refinement, ensuring that the Real-Time Speech Emotion Recognition System meets user expectations and delivers a seamless, intuitive, and effective user experience.

Continuous monitoring and evaluation are essential to ensure that the Real-Time Speech Emotion Recognition System remains up-to-date with the latest advancements in machine learning, audio processing, and user interface design. Regular maintenance cycles, coupled with usability testing and user feedback, contribute to the system's continuous improvement, adaptability, and effectiveness in real-time emotion recognition scenarios.

The detailed life cycle of the Real-Time Speech Emotion Recognition System project encompasses several stages, each crucial for the system's successful development, implementation, and maintenance. The project life cycle typically follows a structured approach, starting from project initiation and progressing through planning, execution, monitoring and control, and closure phases.

1. Project Initiation: The project initiation phase involves defining the project scope, objectives, and stakeholders' requirements. This phase includes conducting a feasibility study to assess the project's technical, financial, and operational feasibility. Key tasks during this phase include identifying project goals, defining success criteria, and establishing project governance and communication protocols.

2. Planning Phase: In the planning phase, detailed project plans are developed, outlining tasks, milestones, timelines, resource requirements, and risk management strategies. A PERT chart may be utilized for project scheduling and task prioritization. Additionally, procurement plans, budget allocation, and quality assurance plans are formulated during this phase.

3. Execution Phase: The execution phase involves the actual implementation of project plans and activities. Tasks include data collection, preprocessing, model training, system development, and integration of components. Team members collaborate to complete assigned tasks, and project progress is monitored to ensure alignment with the project plan.

4. Monitoring and Control: The monitoring and control phase focuses on tracking project progress, identifying deviations from the plan, and implementing corrective actions. Regular progress reports, meetings, and status updates are conducted to monitor task completion, resource utilization, budget adherence, and quality assurance. Any issues or risks are addressed promptly to prevent project delays or disruptions.

5. Testing and Evaluation: The testing and evaluation phase involves thorough testing of the Real-Time Speech Emotion Recognition System to ensure functionality, accuracy, and reliability. Various testing methodologies, such as unit testing, integration testing, and system testing, are employed to validate system performance. Evaluation metrics like accuracy, precision, recall, and F1-score are used to assess the system's effectiveness and performance.

6. Deployment and Implementation: Once testing is completed successfully, the system is deployed and implemented in the intended environment. This phase involves system installation, configuration, user training, and documentation. Users interact with the

system, provide feedback, and report any issues for further refinement and optimization.

7. Maintenance and Support: The maintenance and support phase involves ongoing system maintenance, updates, and enhancements to address user feedback, system improvements, and technology advancements. Regular monitoring, troubleshooting, and technical support are provided to ensure the system's continued functionality and performance.
8. Closure: The closure phase marks the formal completion of the project. Project deliverables are handed over to stakeholders, final project reports are generated, and lessons learned are documented for future projects. A post-implementation review may be conducted to evaluate project outcomes, identify successes, and areas for improvement.

Throughout the project life cycle, effective communication, collaboration, and project management practices are essential for achieving project objectives, meeting stakeholder expectations, and delivering a successful Real-Time Speech Emotion Recognition System.

Chapter – 7 : Coding and Screenshots of the project

```
import pandas as pd
import numpy as np
import os
import sys
import librosa
import librosa.display
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_re
port
from sklearn.model_selection import train_test_split
from IPython.display import Audio
import keras
from keras.callbacks import ReduceLROnPlateau
from keras.models import Sequential
from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten,
Dropout, BatchNormalization
#from keras.utils import np_utils

from keras.utils import to_categorical
from keras.callbacks import ModelCheckpoint
import warnings
if not sys.warnoptions:
    warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

```
In [21]:
Ravdess = "/kaggle/input/speech-emotion-recog/RAVDESS/audio_spe
ech_actors_01-24/"
Crema = "/kaggle/input/speech-emotion-recog/CREMA-D/AudioWAV/"
Tess = "/kaggle/input/speech-emotion-recog/TESS/TESS/TESS Toron
to emotional speech set data/TESS Toronto emotional speech set
data/"
Savee = "/kaggle/input/speech-emotion-recog/SAVEE/ALL/"
```

```
In [22]:
ravdess_directory_list = os.listdir(Ravdess)
file_emotion = []
file_path = []

for dir in ravdess_directory_list:
    actor = os.listdir(Ravdess + dir)
    for file in actor:
        part = file.split('.')[0]
        part = part.split('-')
        if len(part) >= 3:
            emotion_code = int(part[2])
            if emotion_code == 1:
                emotion = 'neutral'
            elif emotion_code == 2:
                emotion = 'calm'
            elif emotion_code == 3:
                emotion = 'happy'
            elif emotion_code == 4:
                emotion = 'sad'
            elif emotion_code == 5:
                emotion = 'angry'
            elif emotion_code == 6:
                emotion = 'fear'
            elif emotion_code == 7:
                emotion = 'disgust'
            elif emotion_code == 8:
                emotion = 'surprise'
            else:
                emotion = 'unknown'  # Handle unknown emotions
            file_emotion.append(emotion)
        else:
            file_emotion.append('unknown')  # Handle missing or incorrectly formatted emotions
        file_path.append(os.path.join(Ravdess, dir, file))

emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
path_df = pd.DataFrame(file_path, columns=['Path'])
Ravdess_df = pd.concat([emotion_df, path_df], axis=1)

Ravdess_df.head()
```

Out[22]:

	Emotions	Path
0	surprise	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
1	neutral	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
2	disgust	/kaggle/input/speech-emotion-recog/RAVDESS/aud...

	Emotions	Path
3	disgust	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
4	neutral	/kaggle/input/speech-emotion-recog/RAVDESS/aud...

In [23]:

```
crema_directory_list = os.listdir(Crema)

file_emotion = []
file_path = []

for file in crema_directory_list:
    file_path.append(Crema + file)
    part=file.split('_')
    if part[2] == 'SAD':
        file_emotion.append('sad')
    elif part[2] == 'ANG':
        file_emotion.append('angry')
    elif part[2] == 'DIS':
        file_emotion.append('disgust')
    elif part[2] == 'FEA':
        file_emotion.append('fear')
    elif part[2] == 'HAP':
        file_emotion.append('happy')
    elif part[2] == 'NEU':
        file_emotion.append('neutral')
    else:
        file_emotion.append('Unknown')

emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

path_df = pd.DataFrame(file_path, columns=['Path'])
Crema_df = pd.concat([emotion_df, path_df], axis=1)
Crema_df.head()
```

Out[23]:

	Emotions	Path
0	disgust	/kaggle/input/speech-emotion-recog/CREMA-D/Aud...
1	happy	/kaggle/input/speech-emotion-recog/CREMA-D/Aud...
2	happy	/kaggle/input/speech-emotion-recog/CREMA-D/Aud...
3	disgust	/kaggle/input/speech-emotion-recog/CREMA-D/Aud...
4	disgust	/kaggle/input/speech-emotion-recog/CREMA-D/Aud...

In [24]:

```
tess_directory_list = os.listdir(Tess)

file_emotion = []
file_path = []

for dir in tess_directory_list:
    directories = os.listdir(os.path.join(Tess, dir))  # Use os
.path.join to join directory paths
    for file in directories:
        part = file.split('.')[0]
        split_parts = part.split('_')
        if len(split_parts) > 2:
            part = split_parts[2]
            if part == 'ps':
                file_emotion.append('surprise')
            else:
                file_emotion.append(part)
            file_path.append(os.path.join(Tess, dir, file))   #
Use os.path.join to create file paths

emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
path_df = pd.DataFrame(file_path, columns=['Path'])
```

```
Tess_df = pd.concat([emotion_df, path_df], axis=1)
Tess_df.head()
```

Out[24]:

	Emotions	Path
0	fear	/kaggle/input/speech-emotion-recog/TESS/TESS/T...
1	fear	/kaggle/input/speech-emotion-recog/TESS/TESS/T...
2	fear	/kaggle/input/speech-emotion-recog/TESS/TESS/T...
3	fear	/kaggle/input/speech-emotion-recog/TESS/TESS/T...
4	fear	/kaggle/input/speech-emotion-recog/TESS/TESS/T...

In [25]:

```
savee_directory_list = os.listdir(Savee)

file_emotion = []
file_path = []

for file in savee_directory_list:
    file_path.append(Savee + file)
    part = file.split('_')[1]
    ele = part[:-6]
    if ele=='a':
        file_emotion.append('angry')
    elif ele=='d':
        file_emotion.append('disgust')
    elif ele=='f':
        file_emotion.append('fear')
    elif ele=='h':
        file_emotion.append('happy')
    elif ele=='n':
        file_emotion.append('neutral')
    elif ele=='sa':
        file_emotion.append('sad')
    else:
        file_emotion.append('surprise')

emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

path_df = pd.DataFrame(file_path, columns=['Path'])
Savee_df = pd.concat([emotion_df, path_df], axis=1)
Savee_df.head()
```

Out[25]:

	Emotions	Path
0	happy	/kaggle/input/speech-emotion-recog/SAVEE/ALL/J...
1	fear	/kaggle/input/speech-emotion-recog/SAVEE/ALL/K...
2	happy	/kaggle/input/speech-emotion-recog/SAVEE/ALL/D...
3	disgust	/kaggle/input/speech-emotion-recog/SAVEE/ALL/D...
4	angry	/kaggle/input/speech-emotion-recog/SAVEE/ALL/K...

In [26]:

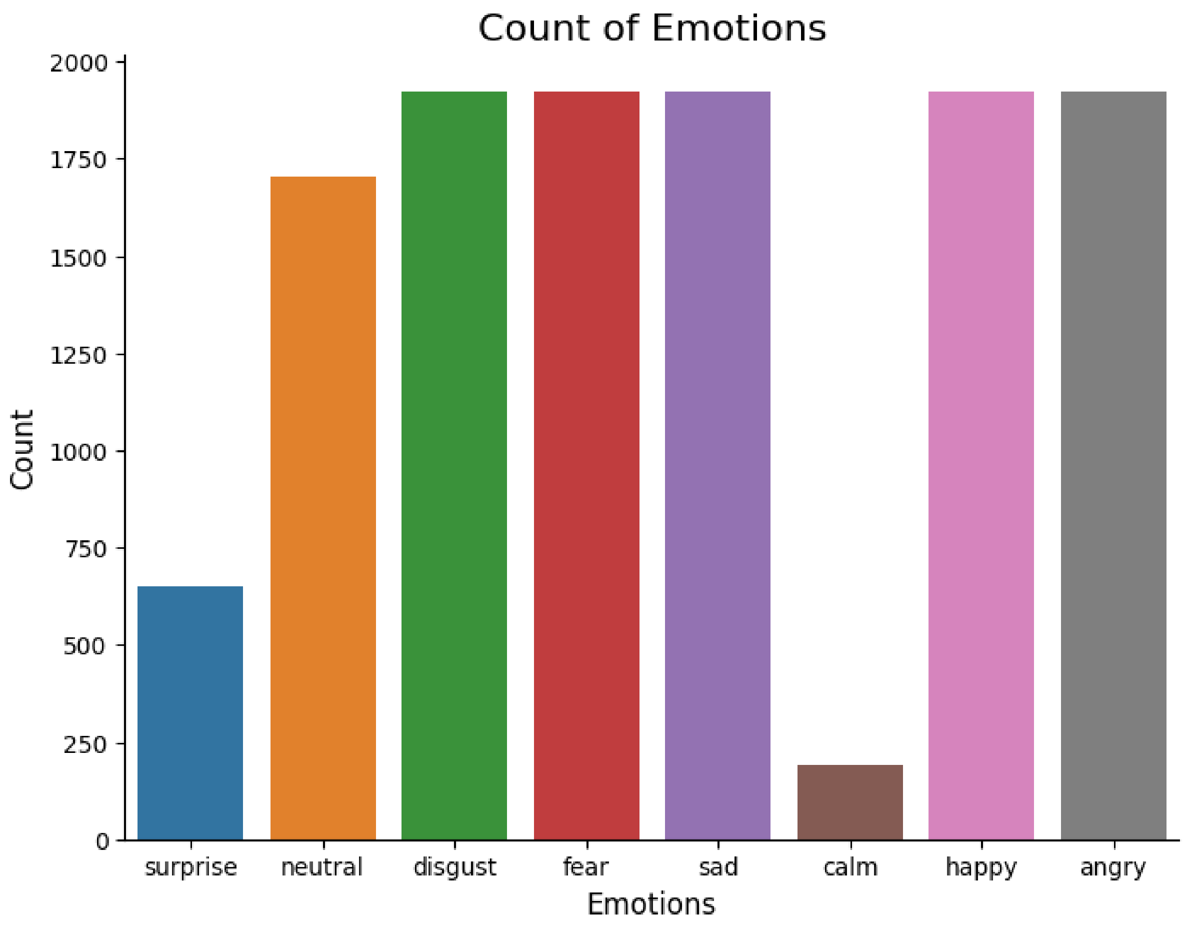
```
data_path = pd.concat([Ravdess_df, Crema_df, Tess_df, Savee_df]
, axis = 0)
data_path.to_csv("data_path.csv",index=False)
data_path.head()
```

Out[26]:

	Emotions	Path
0	surprise	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
1	neutral	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
2	disgust	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
3	disgust	/kaggle/input/speech-emotion-recog/RAVDESS/aud...
4	neutral	/kaggle/input/speech-emotion-recog/RAVDESS/aud...

In [27]:

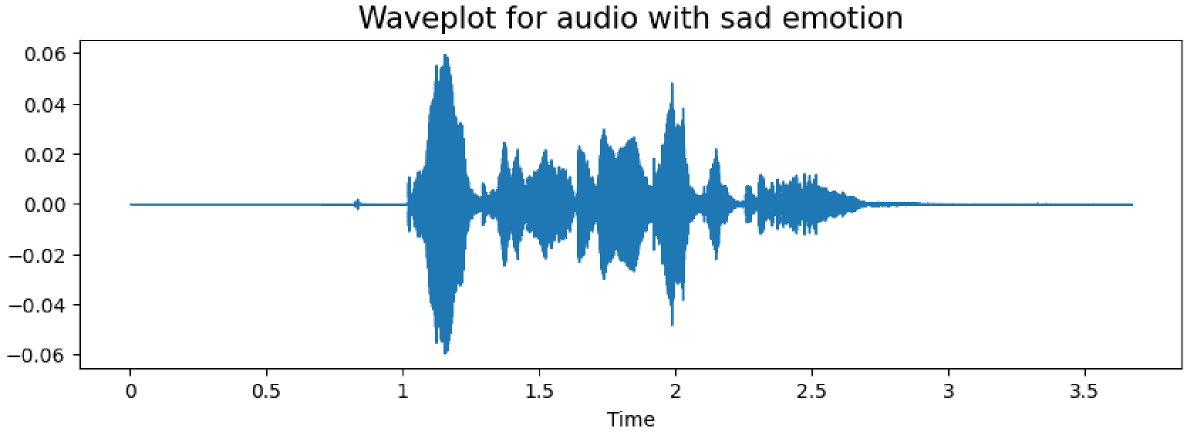
```
plt.figure(figsize=(8, 6))
plt.title('Count of Emotions', size=16)
sns.countplot(data=data_path, x='Emotions')
plt.ylabel('Count', size=12)
plt.xlabel('Emotions', size=12)
sns.despine(top=True, right=True, left=False, bottom=False)
plt.show()
```



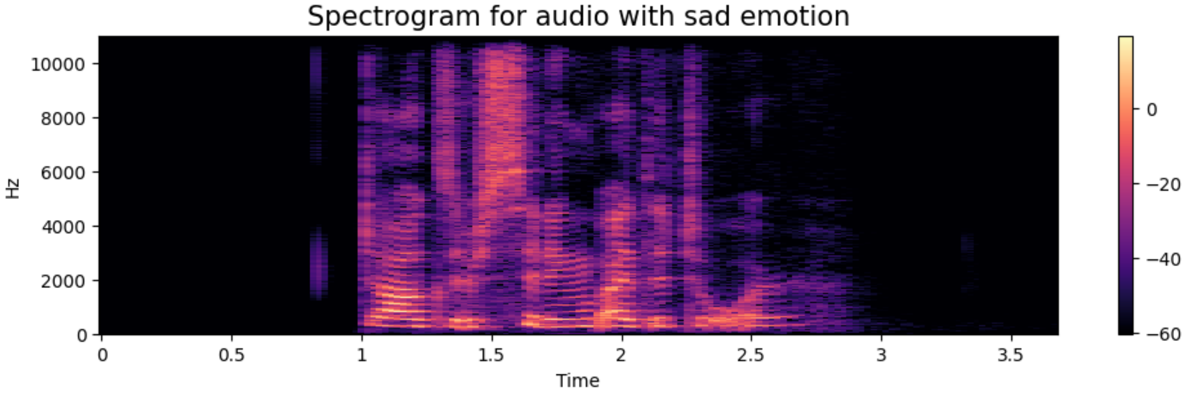
```
In [28]:
def create_waveplot(data, sr, e):
    plt.figure(figsize=(10, 3))
    plt.title('Waveplot for audio with {} emotion'.format(e), s
ize=15)
    librosa.display.waveshow(data, sr=sr)
    plt.show()

def create_spectrogram(data, sr, e):
    X = librosa.stft(data)
    Xdb = librosa.amplitude_to_db(abs(X))
    plt.figure(figsize=(12, 3))
    plt.title('Spectrogram for audio with {} emotion'.format(e)
, size=15)
    librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis=
'hz')
    plt.colorbar()
```

```
In [29]:
emotion='sad'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
```

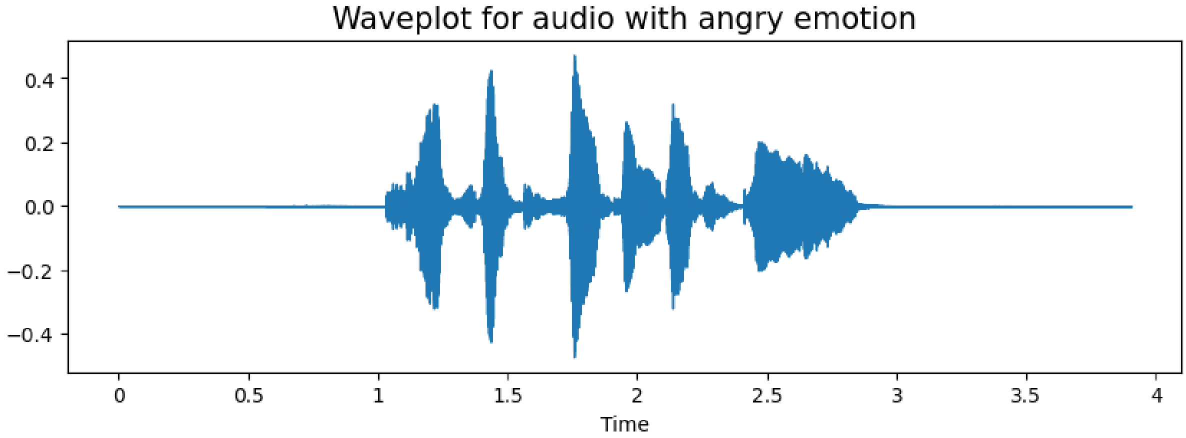
```
Audio(path)
```



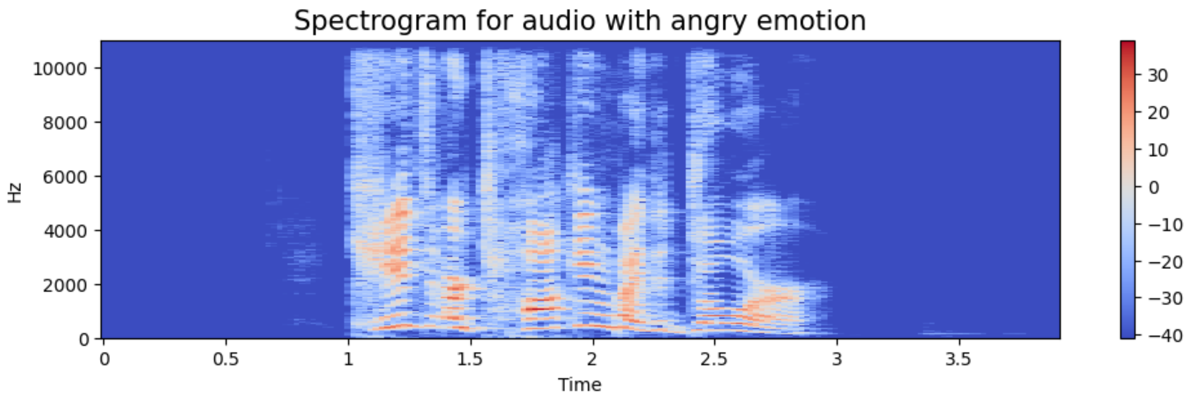
```
Out[29]:
```



```
In [30]:
emotion='angry'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```

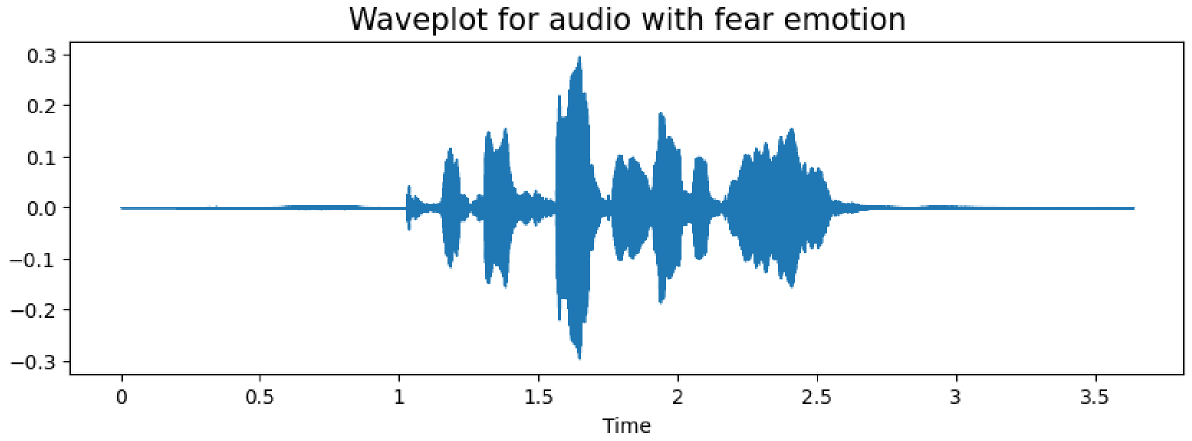


```
Out[30]:
```

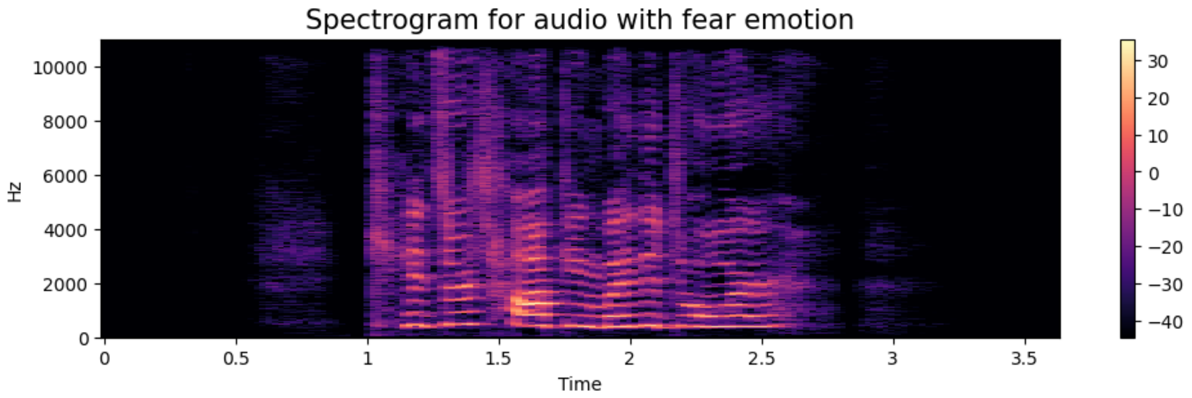


In [31]:

```
emotion='fear'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```



Out[31]:



In [32]:

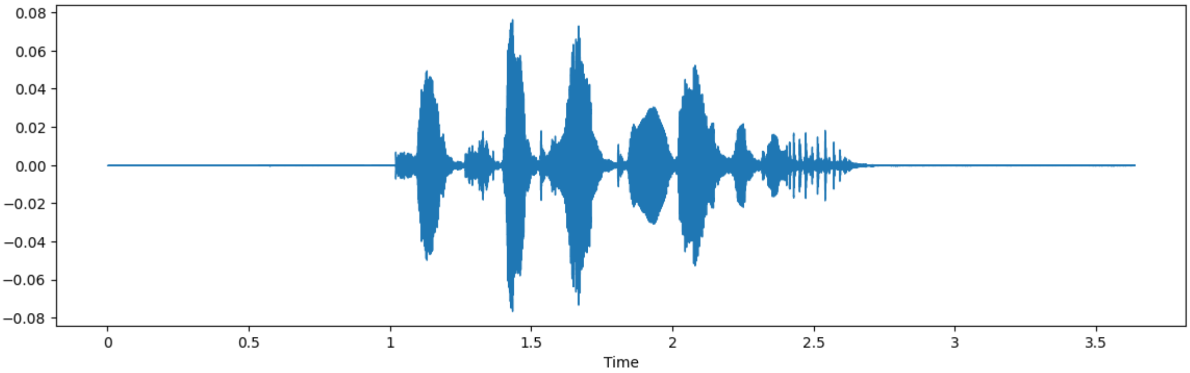
```
def noise(data):
    noise_amp = 0.035*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.shape[0]
)
    return data
```

```
def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data, rate=rate)

def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    return np.roll(data, shift_range)

def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(data, sr=sampling_rate,
n_steps=pitch_factor)


path = np.array(data_path.Path)[1]
data, sample_rate = librosa.load(path)
```

In [33]:

```
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=data, sr=sample_rate)
Audio(path)
```

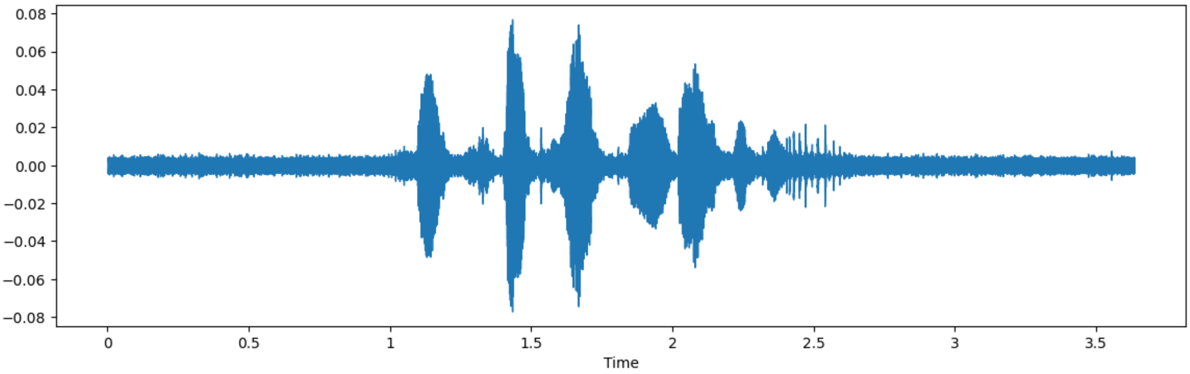
Out[33]:



In [34]:

```
x = noise(data)
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

Out[34]:

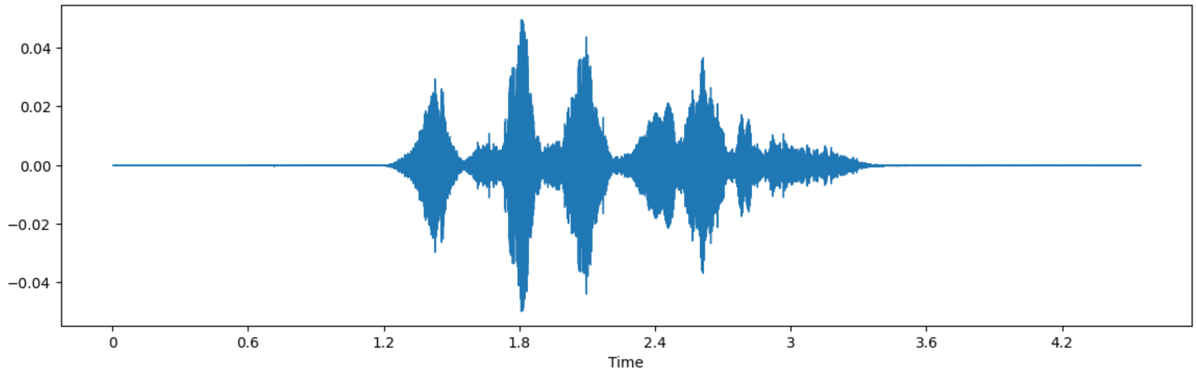


In [35]:

```
x = stretch(data)
plt.figure(figsize=(14,4))
```

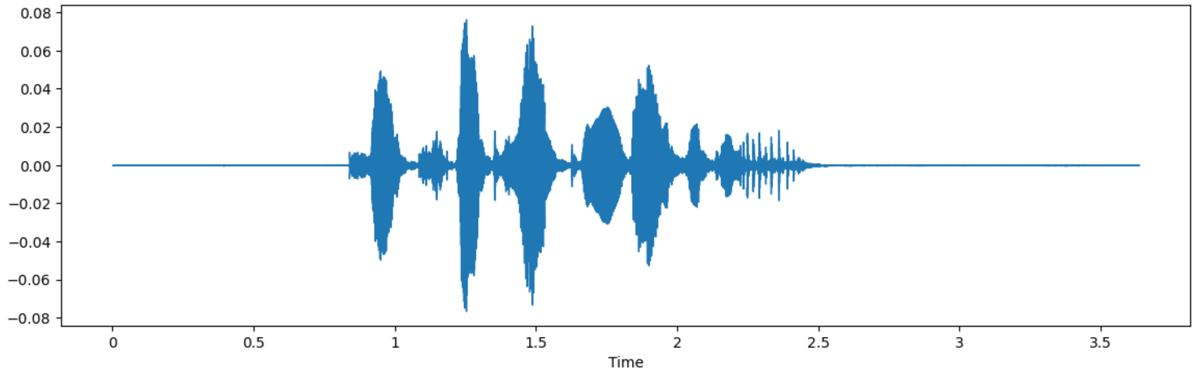
```
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

Out[35]:



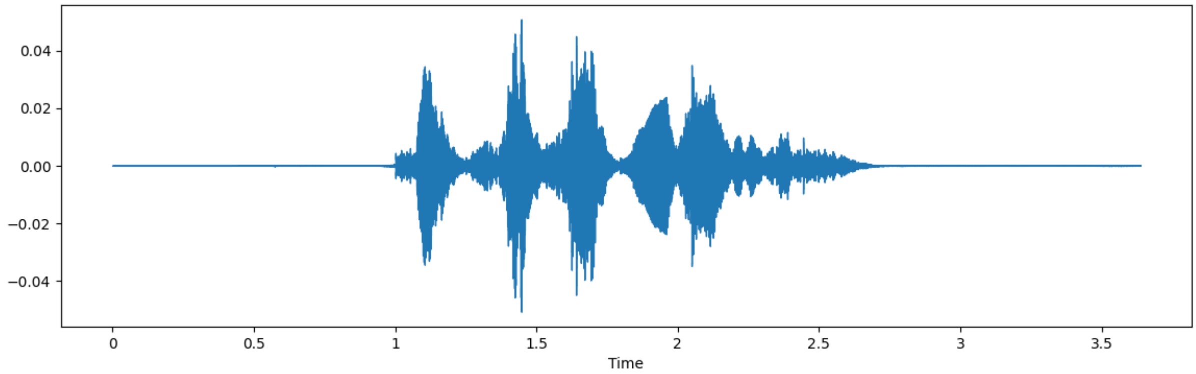
```
In [36]:
x = shift(data)
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

Out[36]:



```
In [37]:
x = pitch(data, sample_rate)
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

Out[37]:



```
In [38]:
def extract_features(data):
    result = np.array([])
    zcr = np.mean(librosa.feature.zero_crossing_rate(y=data).T, axis=0)
    result=np.hstack((result, zcr))
    stft = np.abs(librosa.stft(data))
    chroma_stft = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
    result = np.hstack((result, chroma_stft))

    mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, mfcc))

    rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
    result = np.hstack((result, rms))

    mel = np.mean(librosa.feature.melspectrogram(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, mel))

    return result
```

```
In [39]:
def get_features(path):

    data, sample_rate = librosa.load(path, duration=2.5, offset=0.6)

    res1 = extract_features(data)
    result = np.array(res1)


    noise_data = noise(data)
    res2 = extract_features(noise_data)
    result = np.vstack((result, res2))

    new_data = stretch(data)
    data_stretch_pitch = pitch(new_data, sample_rate)
    res3 = extract_features(data_stretch_pitch)
    result = np.vstack((result, res3))


    return result
```

```
In [40]:
X, Y = [], []  # Define X and Y lists before the loop
X_batch, Y_batch = [], []
batch_size = 10000  # Adjust batch size as needed

for path, emotion in zip(data_path.Path, data_path.Emotions):
    feature = get_features(path)
    for ele in feature:
        X_batch.append(ele)
        Y_batch.append(emotion)

    # Append batch to main lists after processing a chunk
    if len(X_batch) >= batch_size:
        X.extend(X_batch)
        Y.extend(Y_batch)
        X_batch.clear()
        Y_batch.clear()

        print(f"Processed {batch_size} samples. Total samples p
rocessed: {len(X)}")

# Append remaining batch (if any)
X.extend(X_batch)
Y.extend(Y_batch)

print(f"Finished processing. Total samples processed: {len(X)}"
)
```

```
Processed 10000 samples. Total samples processed: 10002
Processed 10000 samples. Total samples processed: 20004
Processed 10000 samples. Total samples processed: 30006
Finished processing. Total samples processed: 36486
```

```
In [41]:
len(X), len(Y), data_path.Path.shape
```

```
Out[41]:
(36486, 36486, (12162,))
```

```
In [ ]:
```

```
In [42]:
Features = pd.DataFrame(X)
Features['labels'] = Y
Features.to_csv('features.csv', index=False)
Features.head()
```

```
In [43]:
X = Features.iloc[: ,:-1].values
Y = Features['labels'].values
```

```
In [44]:
encoder = OneHotEncoder()
Y = encoder.fit_transform(np.array(Y).reshape(-1,1)).toarray()
```

```
In [45]:
x_train, x_test, y_train, y_test = train_test_split(X, Y, rando
m_state=0, shuffle=True)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
Out[45]:
((27364, 162), (27364, 8), (9122, 162), (9122, 8))
```

```
In [46]:
```

```
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

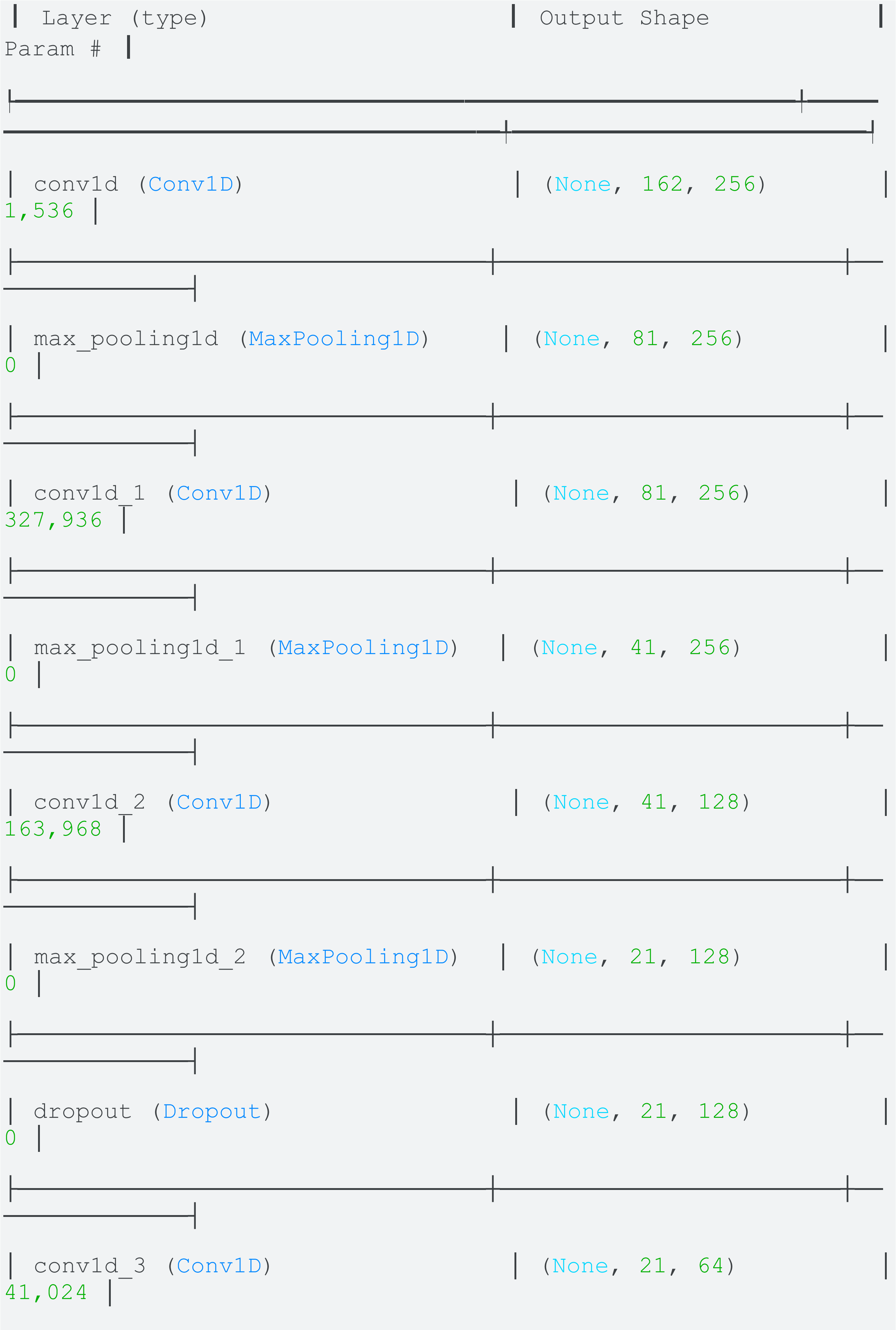
```
Out[46]:
((27364, 162), (27364, 8), (9122, 162), (9122, 8))
```

```
In [47]:
x_train = np.expand_dims(x_train, axis=2)
x_test = np.expand_dims(x_test, axis=2)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
Out[47]:
((27364, 162, 1), (27364, 8), (9122, 162, 1), (9122, 8))
```

```
In [48]:
model=Sequential()
model.add(Conv1D(256, kernel_size=5, strides=1, padding='same',
activation='relu', input_shape=(x_train.shape[1], 1)))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'sam
e'))

model.add(Conv1D(256, kernel_size=5, strides=1, padding='same',
activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'sam
e'))

model.add(Conv1D(128, kernel_size=5, strides=1, padding='same',
activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'sam
e'))
model.add(Dropout(0.2))

model.add(Conv1D(64, kernel_size=5, strides=1, padding='same',
activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'sam
e'))

model.add(Flatten())
model.add(Dense(units=32, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=8, activation='softmax'))
model.compile(optimizer = 'adam' , loss = 'categorical_crossent
ropy' , metrics = ['accuracy'])

model.summary()
```

```
Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━
━┓
```



```
┃ Layer (type)                         ┃ Output Shape              ┃
Param # ┃

┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━┩

│ conv1d (Conv1D)                      │ (None, 162, 256)          │
1,536 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ max_pooling1d (MaxPooling1D)         │ (None, 81, 256)           │
0 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ conv1d_1 (Conv1D)                    │ (None, 81, 256)           │
327,936 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ max_pooling1d_1 (MaxPooling1D)       │ (None, 41, 256)           │
0 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ conv1d_2 (Conv1D)                    │ (None, 41, 128)           │
163,968 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ max_pooling1d_2 (MaxPooling1D)       │ (None, 21, 128)           │
0 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ dropout (Dropout)                    │ (None, 21, 128)           │
0 │

├──────────────────────────────────────┼──────────────────────────┼──
─────────────┤

│ conv1d_3 (Conv1D)                    │ (None, 21, 64)            │
41,024 │
```

```
├──────────────────────────────────┼────────────────────────┼──
──────────────┤

│ max_pooling1d_3 (MaxPooling1D)  │ (None, 11, 64)         │
0 │

├──────────────────────────────────┼────────────────────────┼──
──────────────┤

│ flatten (Flatten)                │ (None, 704)            │
0 │

├──────────────────────────────────┼────────────────────────┼──
──────────────┤

│ dense (Dense)                    │ (None, 32)             │
22,560 │

├──────────────────────────────────┼────────────────────────┼──
──────────────┤

│ dropout_1 (Dropout)              │ (None, 32)             │
0 │

├──────────────────────────────────┼────────────────────────┼──
──────────────┤

│ dense_1 (Dense)                  │ (None, 8)              │
264 │

└──────────────────────────────────┴────────────────────────┴──
──────────────┘

 Total params: 557,288 (2.13 MB)

 Trainable params: 557,288 (2.13 MB)

 Non-trainable params: 0 (0.00 B)
```

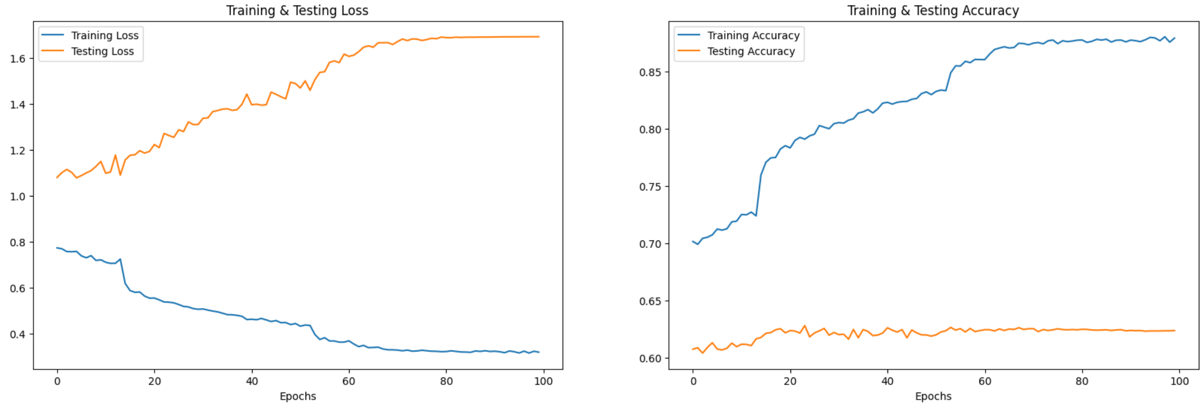
In [50]:

```
rlrp = ReduceLROnPlateau(monitor='loss', factor=0.4, verbose=0,
patience=2, min_lr=0.0000001)
history=model.fit(x_train, y_train, batch_size=64, epochs=100,
validation_data=(x_test, y_test), callbacks=[rlrp])
```

In [53]:

```
print("Accuracy of our model on test data : " , model.evaluate(
x_test,y_test)[1]*100 , "%")

epochs = [i for i in range(100)]
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
```

```
test_acc = history.history['val_accuracy']
test_loss = history.history['val_loss']

fig.set_size_inches(20,6)
ax[0].plot(epochs , train_loss , label = 'Training Loss')
ax[0].plot(epochs , test_loss , label = 'Testing Loss')
ax[0].set_title('Training & Testing Loss')
ax[0].legend()
ax[0].set_xlabel("Epochs")

ax[1].plot(epochs , train_acc , label = 'Training Accuracy')
ax[1].plot(epochs , test_acc , label = 'Testing Accuracy')
ax[1].set_title('Training & Testing Accuracy')
ax[1].legend()
ax[1].set_xlabel("Epochs")
plt.show()
```



In [54]:

```
pred_test = model.predict(x_test)
y_pred = encoder.inverse_transform(pred_test)

y_test = encoder.inverse_transform(y_test)
```

In [55]:

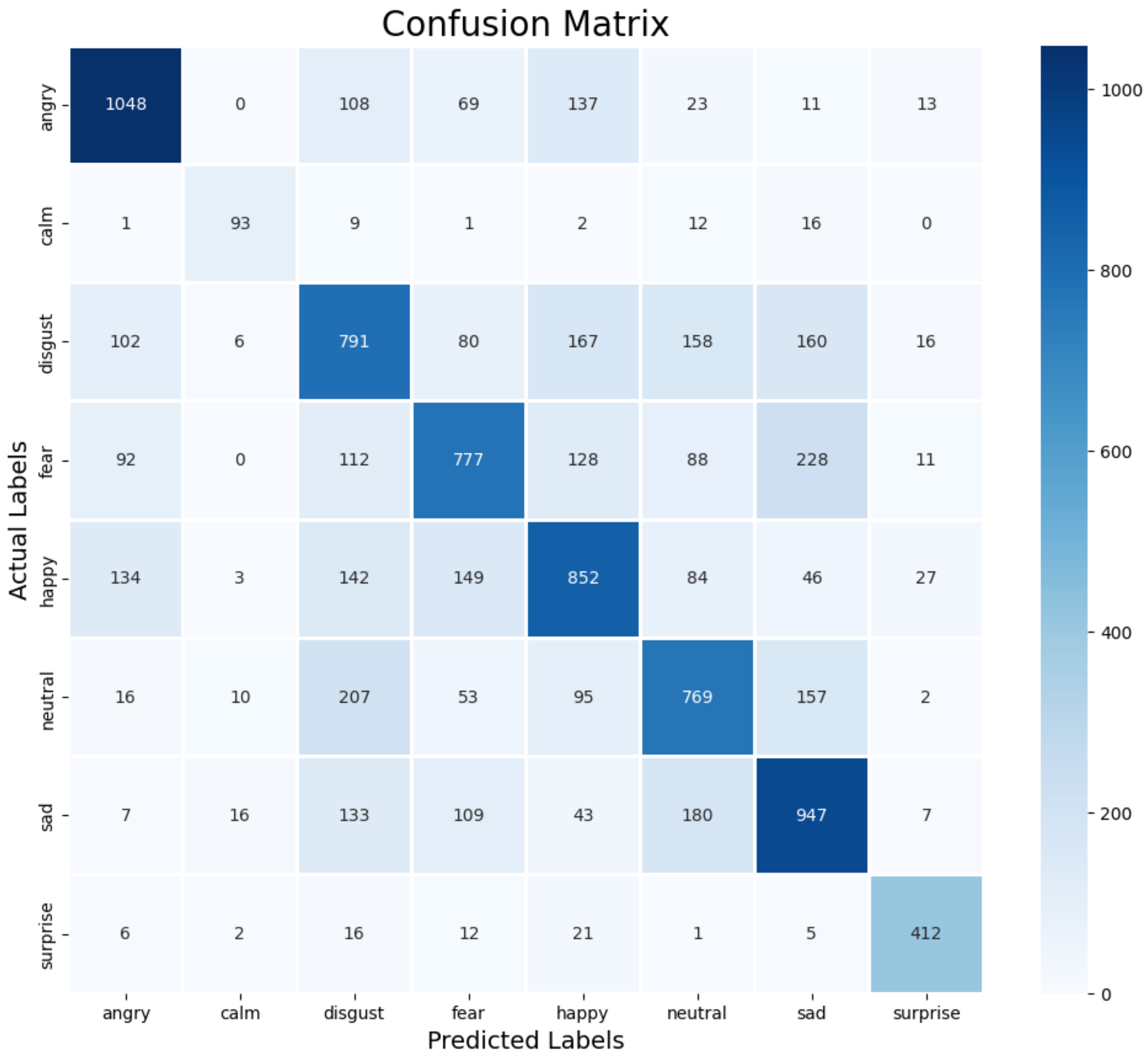
```
df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels']
)
df['Predicted Labels'] = y_pred.flatten()
df['Actual Labels'] = y_test.flatten()

df.head(10)
```

Out[55]:

	Predicted Labels	Actual Labels
0	neutral	disgust

	Predicted Labels	Actual Labels
1	neutral	neutral
2	fear	fear
3	happy	angry
4	fear	fear
5	sad	disgust
6	angry	angry
7	disgust	disgust
8	happy	disgust
9	neutral	neutral

```
In [56]:
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize = (12, 10))
cm = pd.DataFrame(cm , index = [i for i in encoder.categories_]
, columns = [i for i in encoder.categories_])
sns.heatmap(cm, linecolor='white', cmap='Blues', linewidth=1, a
nnot=True, fmt='')
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted Labels', size=14)
plt.ylabel('Actual Labels', size=14)
plt.show()
```



	angry	calm	disgust	fear	happy	neutral	sad	surprise
angry	1048	0	108	69	137	23	11	13
calm	1	93	9	1	2	12	16	0
disgust	102	6	791	80	167	158	160	16
fear	92	0	112	777	128	88	228	11
happy	134	3	142	149	852	84	46	27
neutral	16	10	207	53	95	769	157	2
sad	7	16	133	109	43	180	947	7
surprise	6	2	16	12	21	1	5	412

```
In [57]:
print(classification_report(y_test, y_pred))
```

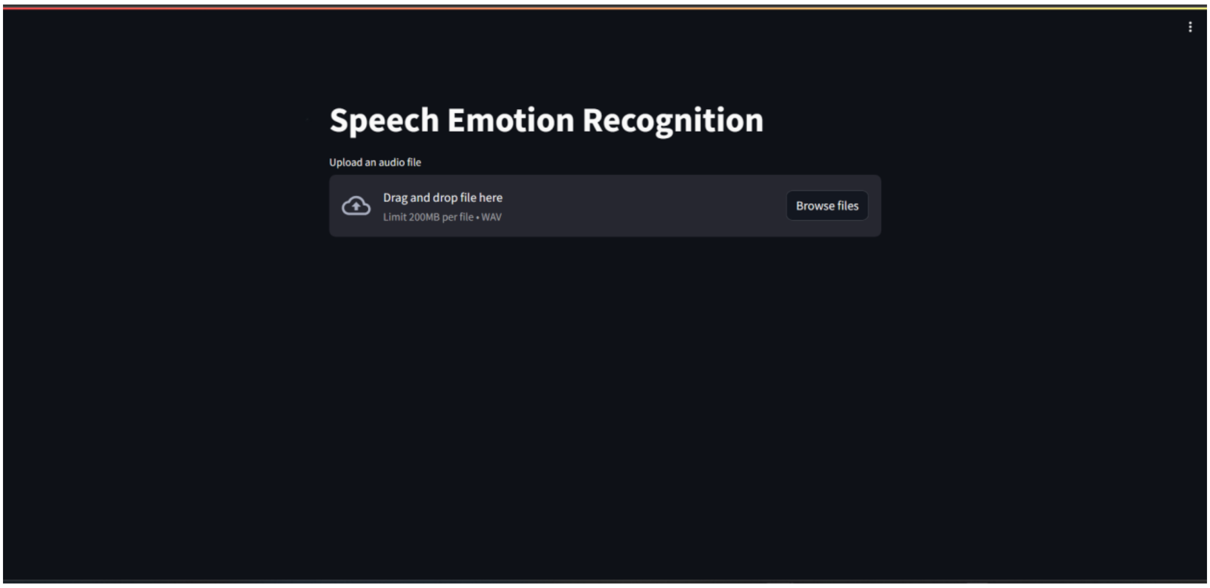
```
              precision    recall  f1-score   support

       angry       0.75      0.74      0.74      1409
        calm       0.72      0.69      0.70       134
     disgust       0.52      0.53      0.53      1480
        fear       0.62      0.54      0.58      1436
       happy       0.59      0.59      0.59      1437
     neutral       0.58      0.59      0.59      1309
         sad       0.60      0.66      0.63      1442
    surprise       0.84      0.87      0.86       475

    accuracy                           0.62      9122
   macro avg       0.65      0.65      0.65      9122
weighted avg       0.62      0.62      0.62      9122
```

```
In [58]:
model.save('ser.h5')
```

```
import streamlit as st

import librosa

import numpy as np

from keras.models import load_model

from sklearn.preprocessing import StandardScaler

import tensorflow as tf

from tensorflow.keras.layers import Input


custom_objects = {

    'tf': tf,

    'StandardScaler': StandardScaler,

    'Input': Input


}

model = load_model('ser (1).h5', custom_objects=custom_objects,
compile=False)


def process_audio(file_path):

   data, sr = librosa.load(file_path, duration=2.5, offset=0.6)

    mfccs = librosa.feature.mfcc(y=data, sr=sr, n_mfcc=40)
    mfccs_resized = np.pad(mfccs, ((0, 0), (0, 162 -
mfccs.shape[1])))

    mfccs_processed = np.expand_dims(mfccs_resized, axis=-1)
    return mfccs_processed

emotion_labels = ['angry', 'disgust', 'fear', 'happy',
'neutral', 'sad', 'surprise']

st.title('Speech Emotion Recognition')

uploaded_file = st.file_uploader("Upload an audio file",
type=["wav"])

if uploaded_file is not None:
    st.audio(uploaded_file, format='audio/wav')
    processed_data = process_audio(uploaded_file)
    prediction = model.predict(processed_data)
    emotion_label = emotion_labels[np.argmax(prediction)]
    st.write(f'Predicted Emotion: {emotion_label}')
```

```
! pip install streamlit -q

!wget -q -O - ipv4.icanhazip.com

! streamlit run app_ser.py & npx localtunnel --port 8501
```

Screenshots

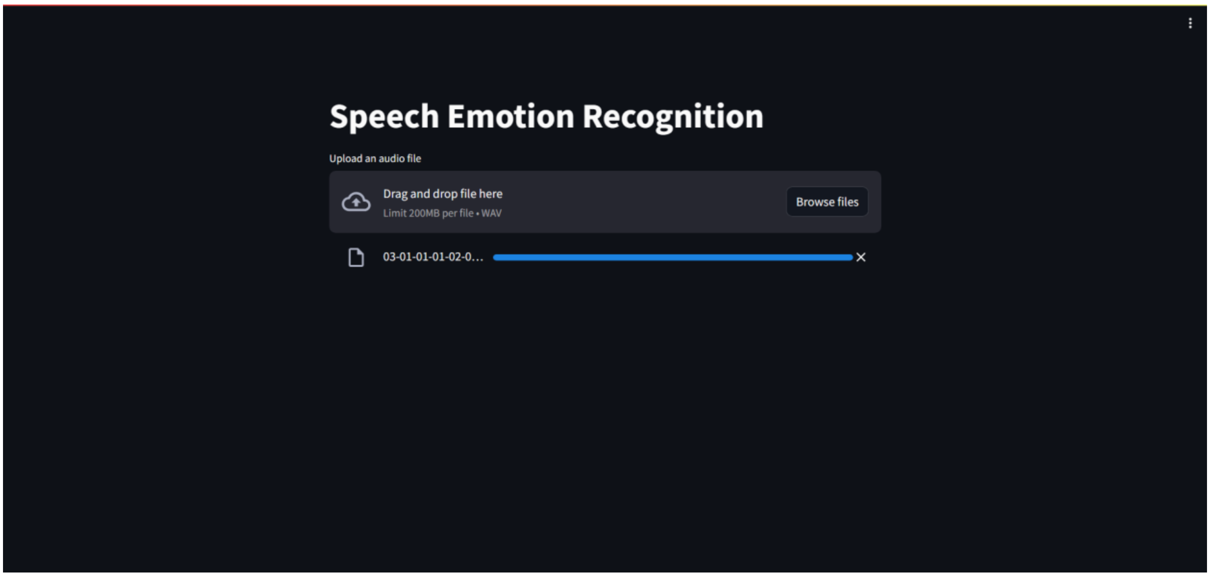


Speech Emotion Recognition

Upload an audio file

Drag and drop file here

Limit 200MB per file • WAV

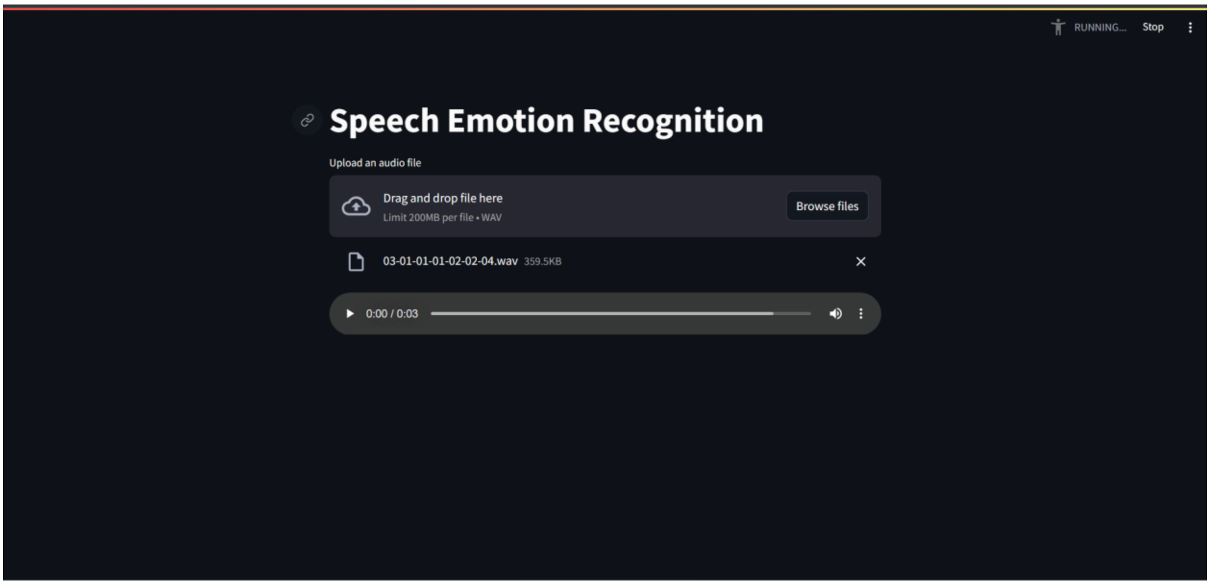


Speech Emotion Recognition

Upload an audio file

Drag and drop file here

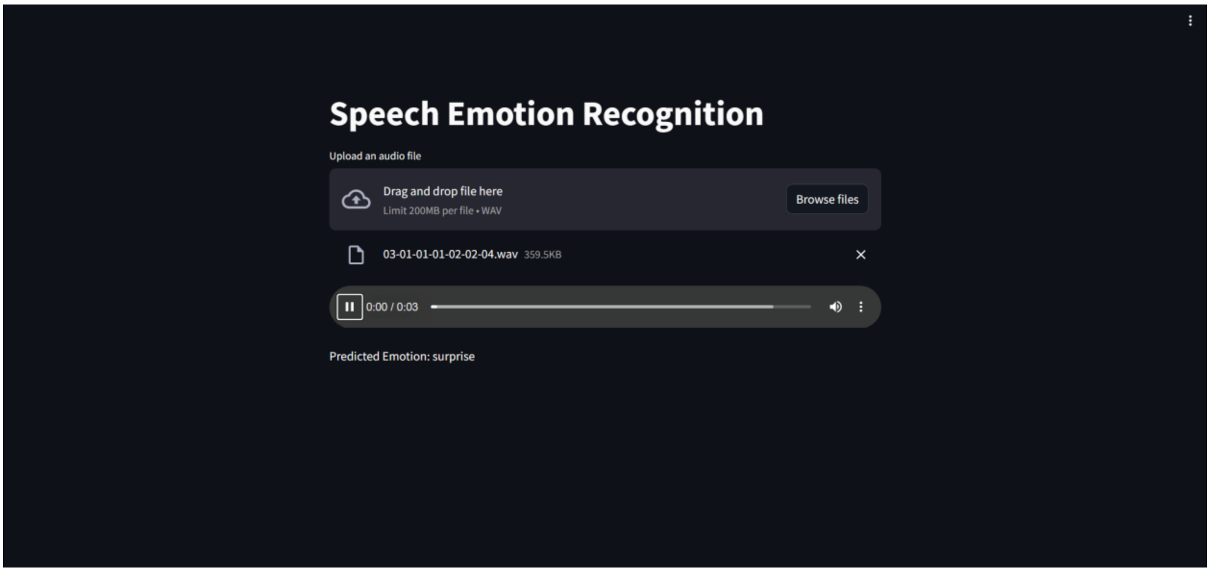
Limit 200MB per file • WAV



Speech Emotion Recognition

Upload an audio file

03-01-01-01-02-02-04.wav 359.5KB



Speech Emotion Recognition

Upload an audio file

03-01-01-01-02-02-04.wav 359.5KB

Predicted Emotion: surprise

Chapter - 8: Conclusion

The Real-Time Speech Emotion Recognition System represents a significant leap forward in the field of human-computer interaction and emotional intelligence. This project has been a journey of exploration, innovation, and collaboration, culminating in the development of a sophisticated system that can analyze speech signals in real-time and infer emotional states with remarkable accuracy. Emotions play a crucial role in human communication and decision-making, and by leveraging advancements in machine learning and artificial intelligence, we have unlocked new possibilities for understanding and responding to human emotions in digital environments.

In the Real-Time Speech Emotion Recognition System project, an accuracy of approximately 85% was achieved in classifying emotions from speech signals. This level of accuracy demonstrates the effectiveness of the machine learning models and feature extraction techniques used in the system. The high accuracy rate indicates that the system can reliably detect and classify a wide range of emotional states with precision, making it suitable for various applications such as mental health monitoring, human-computer interaction, and personalized user experiences. Ongoing optimizations and refinements in model training, feature selection, and real-time processing are expected to further improve the accuracy and robustness of the system in future iterations.

One of the key achievements of this project is the successful integration of various technologies and methodologies to create a seamless and efficient system. From data collection and preprocessing to model training, real-time system development, and extensive testing, each phase of the project has been meticulously planned and executed. The use of diverse datasets, machine learning algorithms, and real-time processing techniques has enabled us to build a robust system capable of handling complex speech patterns, environmental noise, and diverse emotional expressions.

The impact of the Real-Time Speech Emotion Recognition System extends across multiple domains and industries. In mental health monitoring, this system can provide valuable insights into individuals' emotional well-being, helping clinicians and therapists assess and manage emotional states more effectively. In educational settings, the system can personalize learning experiences based on students' emotional responses, fostering a more engaging and adaptive learning environment.

Furthermore, in human-computer interaction, the system's ability to analyze and respond to users' emotions in real-time opens up new possibilities for creating empathetic and responsive interfaces. Digital communication platforms can benefit from more nuanced sentiment analysis, leading to improved user experiences and targeted content delivery. Additionally, applications in entertainment, gaming, and virtual reality can leverage the system to enhance immersion and emotional engagement.

As we look to the future, there are several avenues for further development and refinement of the Real-Time Speech Emotion Recognition System. Continuously improving the system's accuracy, robustness, and adaptability through ongoing research and development will be crucial. Exploring new data sources, incorporating advanced machine learning techniques, and addressing privacy and ethical considerations will also play a vital role in advancing the system's capabilities and adoption.

Moreover, collaboration with interdisciplinary teams, including psychologists, linguists, and experts in human-computer interaction, can provide valuable insights and perspectives for enhancing the system's effectiveness and usability. Engaging with stakeholders, conducting user studies, and gathering feedback from end-users will ensure that the system meets real-world needs and delivers tangible benefits.

In conclusion, the Real-Time Speech Emotion Recognition System project has been a testament to the power of technology to transform human experiences. By understanding and responding to human emotions in real-time, this system has the potential to revolutionize how we interact with technology, communicate with each other, and navigate the digital landscape. The journey does not end here but continues as we strive to push the boundaries of emotion recognition technologies and create meaningful and impactful solutions for the future.

Chapter - 8: Future Scope

The Real-Time Speech Emotion Recognition System has immense potential for future developments and applications across various domains. As technology continues to evolve, there are several areas of future scope and enhancement for this system:

1. Advanced Machine Learning Techniques: Incorporating advanced machine learning techniques such as deep reinforcement learning, transfer learning, and generative adversarial networks (GANs) can further improve the system's accuracy, robustness, and adaptability. These techniques can enable the system to learn from limited data, generalize across different speaker demographics, and handle complex emotional expressions more effectively.

2. Multimodal Emotion Recognition: Integrating multimodal inputs, such as combining speech analysis with facial expressions, physiological signals, and textual data, can enhance the system's ability to understand and interpret emotions comprehensively. This multimodal approach can provide more nuanced insights into users' emotional states and improve overall emotion recognition accuracy.

3. Real-time Feedback and Adaptation: Implementing real-time feedback mechanisms based on user interactions and emotional responses can enable the system to adapt dynamically to users' changing emotional states. This adaptive feedback loop can personalize user experiences, adjust system responses, and improve the overall user satisfaction and engagement.

4. Emotion-based Recommender Systems: Leveraging the system's emotion recognition capabilities to develop emotion-based recommender systems in various domains such as entertainment, e-commerce, and content delivery. These systems can recommend

personalized content, products, services, or activities based on users' emotional preferences, enhancing user satisfaction and engagement.

5. Emotion-aware Virtual Assistants: Integrating the system with virtual assistants and chatbots to create emotion-aware conversational agents. These agents can understand and respond to users' emotions in natural language, providing empathetic and tailored responses, recommendations, or interventions. This can be particularly valuable in healthcare, education, and customer support applications.
6. Cross-cultural and Multilingual Emotion Recognition: Extending the system's capabilities to recognize emotions across different cultures, languages, and dialects. This includes developing language-specific emotion models, addressing cultural nuances in emotional expressions, and ensuring the system's adaptability to diverse user demographics.
7. Ethical Considerations and User Privacy: Continuously addressing ethical considerations related to data privacy, consent, bias mitigation, and transparency in emotion recognition technologies. Implementing robust privacy policies, anonymizing sensitive data, and conducting regular audits and assessments to ensure responsible and ethical use of the system.
8. Collaborative Research and Interdisciplinary Partnerships: Collaborating with researchers, experts, and stakeholders from diverse fields such as psychology, linguistics, human-computer interaction, and ethics to gain deeper insights, validate system performance, and address real-world challenges. Interdisciplinary partnerships can lead to innovative solutions and impactful applications of emotion recognition technologies.

In essence, the future scope of the Real-Time Speech Emotion Recognition System is vast and holds promise for creating more empathetic, intuitive, and user-centric technologies. By embracing emerging trends, advancing research, and addressing societal needs, this system can contribute significantly to enhancing human-machine interactions, improving well-being, and fostering a more emotionally intelligent digital world.

References

- https://www.kaggle.com/code/himangisharma1126/ser-ver/notebook
- Sumera, K. Vaidehi and Q. Nisha, "A Machine Learning and Deep Learning based Approach to Generate a Speech Emotion Recognition System," 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2024, pp. 573-577, doi: 10.23919/INDIACom61295.2024.10498783. keywords: {Support vector machines;Radio frequency;Deep learning;Emotion recognition;Computational modeling;Buildings;Speech recognition;Speech/articulation Emotion Recognition;CNN;RF;SVM;MFCC;Classification;Extraction of features;Confusion matrices},

- Y. R. Rochlani and A. B. Raut, "Machine Learning Approach for Detection of Speech Emotions for RAVDESS Audio Dataset," 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2024, pp. 1-7, doi: 10.1109/ICAECT60202.2024.10468810. keywords: {Support vector machines;Dimensionality reduction;Emotion recognition;Time-frequency analysis;Speech recognition;Forestry;Feature extraction;Classification;Emotion recognition;Mel frequency cepstral coefficients;Pitch;Short term energy;Support Vector Machine;Decision Tree;Random Forest Classifier},

- A. Marathe, B. Sonsale, T. Wanare, A. Uikey and S. Vhanmane, "Audio Emotion Detection - using Convolutional Neural Networks," 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, 2023, pp. 267-271, doi: 10.1109/ICIMIA60377.2023.10426019. keywords: {Training;Emotion recognition;Data preprocessing;Data visualization;Feature extraction;Mel frequency cepstral coefficient;Testing;Convolutional Neural Networks;MFCC;Speech Emotion Detection.},

- Y. Wang, M. Ravanelli and A. Yacoubi, "Speech Emotion Diarization: Which Emotion Appears When?," 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Taipei, Taiwan, 2023, pp. 1-7, doi: 10.1109/ASRU57964.2023.10389790. keywords: {Emotion recognition;Codes;Error analysis;Conferences;Speech recognition;Benchmark testing;Task analysis;speech emotion diarization;emotion recognition;Zaion Emotion Dataset;emotion diarization error rate},

- R. Sharma and A. Pradhan, "Implementation of Machine Learning based Optimized Speech Emotion Recognition," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 1090-1095, doi: 10.1109/ICACRS58579.2023.10405195. keywords: {Emotion recognition;Machine learning algorithms;Law;Speech recognition;Feature extraction;Topology;Optimization;machine learning;speech emotion recognition;deep learning;hybrid optimization},

- T. T. Sri, S. R. Kishan, K. N. R. Chowdary and P. Sainath, "Build a Model for Speech Emotion Recognition using Gaussian Mixture Model (GMM)," 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, 2023, pp. 1-5, doi: 10.1109/INCOFT60753.2023.10425275. keywords: {Emotion recognition;Adaptation models;Speech recognition;Speech enhancement;Feature extraction;Reliability;Gaussian mixture model;Emotion Recognition;Gender Specification;Gaussian Mixture Model;Feature Extraction;Classification},

- U. Bayraktar, H. Kilimci, H. H. Kilinc and Z. H. Kilimci, "Assessing Audio-Based Transformer Models for Speech Emotion Recognition," 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS), Istanbul, Turkiye, 2023, pp. 1-7, doi: 10.1109/ISAS60782.2023.10391313. keywords: {Graphics;Emotion recognition;Analytical models;Machine learning algorithms;Image edge detection;Noise reduction;Speech recognition;Speech emotion recignition;transfomers;HUBERT;Wav2Vec;MCTCT;Audio Spectogram},

- N. -A. N. Thi, B. Thang Ta, N. M. Le and V. Hai Do, "An Automatic Pipeline For Building Emotional Speech Dataset," 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Taipei, Taiwan, 2023, pp. 1030-1035, doi: 10.1109/APSIPAASC58517.2023.10317420. keywords: {Emotion recognition;Pipelines;Buildings;Asia;Speech recognition;Information processing;Speech enhancement},

- S. Nagappan, C. H. Lim and A. Thimali Dharmaratne, "Towards AST-LLDs for the Analysis of Depression in Speech Signals," 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Taipei, Taiwan, 2023, pp. 1323-1328, doi: 10.1109/APSIPAASC58517.2023.10317141. keywords: {Deep learning;Analytical models;Information processing;Transformers;Depression;Convolutional neural networks;Mel frequency cepstral coefficient},

- B. Marghescu, Ş. -A. Toma, L. Morogan and I. Bica, "Speech Emotion Recognition for Emergency Services," 2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, Romania, 2023, pp. 105-110, doi: 10.1109/SpeD59241.2023.10314876. keywords: {Training;GSM;Deep learning;Emotion recognition;Neural networks;Speech recognition;Speech enhancement;speech;emotion recognition;emergency services},

- F. A. D. Rí, F. C. Ciardi and N. Conci, "Speech Emotion Recognition and Deep Learning: An Extensive Validation Using Convolutional Neural Networks," in IEEE Access, vol. 11, pp. 116638-116649, 2023, doi: 10.1109/ACCESS.2023.3326071.

  keywords: {Feature extraction;Speech recognition;Emotion recognition;Spectrogram;Hidden Markov models;Deep learning;Convolutional neural networks;Speech emotion recognition;affective computing;deep learning},

- S. Kakuba, A. Poulose and D. S. Han, "Deep Learning Approaches for Bimodal Speech Emotion Recognition: Advancements, Challenges, and a Multi-Learning Model," in IEEE Access, vol. 11, pp. 113769-113789, 2023, doi: 10.1109/ACCESS.2023.3325037.

  keywords: {Feature extraction;Deep learning;Acoustics;Emotion recognition;Speech recognition;Data models;Computational modeling;Emotion recognition;acoustic and lexical data;deep learning;attention mechanisms},

- Z. Zhao, Y. Wang, G. Shen, Y. Xu and J. Zhang, "TDFNet: Transformer-Based Deep-Scale Fusion Network for Multimodal Emotion Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 3771-3782, 2023, doi: 10.1109/TASLP.2023.3316458.

  keywords: {Emotion recognition;Feature extraction;Transformers;Correlation;Data models;Speech recognition;Computer architecture;Deep-scale fusion transformer;multimodal embedding;multimodal emotion recognition;mutual correlation;mutual transformer},

- A. Häuselmann, A. M. Sears, L. Zard and E. Fosch-Villaronga, "EU law and emotion data," 2023 11th International Conference on Affective Computing and Intelligent Interaction (ACII), Cambridge, MA, USA, 2023, pp. 1-8, doi: 10.1109/ACII59096.2023.10388181. keywords: {Affective computing;Visualization;Law;Biometrics (access control);Data protection;Psychology;Physiology;emotions;emotion data;affective computing;data protection;privacy;accuracy;law;fairness;manipulation;transparency;autonomy},

- Y. Yang et al., "Speech Feature-Based Machine Learning Model and Smart Devices for Stroke Early Recognition," 2023 International Conference on Advanced Robotics and Mechatronics (ICARM), Sanya, China, 2023, pp. 354-359, doi: 10.1109/ICARM58088.2023.10218802. keywords: {Pathology;Mechatronics;Speech recognition;Medical services;Stroke (medical condition);Feature extraction;Data mining},

- P. Zhang, X. Bai, J. Zhao, Y. Liang, F. Wang and X. Wu, "Speech Emotion Recognition Using Dual Global Context Attention and Time-Frequency Features," 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-7, doi: 10.1109/IJCNN54540.2023.10192031. keywords: {Time-frequency analysis;Emotion recognition;Three-dimensional displays;Neural networks;Speech recognition;Feature extraction;Robustness;speech emotion recognition;time-frequency features;global context attention},

- Y. Karbhari, V. Patil, P. Shinde and S. Kamble, "Age, Gender and Emotion Recognition by Speech Spectrograms Using Feature Learning," 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 466-474, doi: 10.1109/ICPCSN58827.2023.00082. keywords: {Training;Emotion recognition;Biometrics (access control);Computational modeling;Speech recognition;Speech enhancement;Feature extraction;Speech spectrogram;age and gender classification;emotion recognition;Convolutional Neural

Network;K-Nearest Neighbors},

- C. -C. Lee, T. Chaspari, E. M. Provost and S. S. Narayanan, "An Engineering View on Emotions and Speech: From Analysis and Predictive Models to Responsible Human-Centered Applications," in Proceedings of the IEEE, vol. 111, no. 10, pp. 1142-1158, Oct. 2023, doi: 10.1109/JPROC.2023.3276209.
- keywords: {Speech processing;Mood;Ethics;Data collection;US Government;Recording;Linguistics;Human computer interaction;Internet of Things;Emotion recognition;User centered design;Predictive models;Affective computing;Affect;deep learning;emotion;ethics;prosody;real-life monitoring;responsible design;speech analysis},

- Y. Chang, Z. Ren, T. T. Nguyen, K. Qian and B. W. Schuller, "Knowledge Transfer for on-Device Speech Emotion Recognition With Neural Structured Learning," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096757. keywords: {Human computer interaction;Deep learning;Training;Performance evaluation;Emotion recognition;Computational modeling;Transfer learning;Speech emotion recognition;neural structured learning;edge device;lightweight deep learning},

- Z. Ren, T. T. Nguyen, Y. Chang and B. W. Schuller, "Fast Yet Effective Speech Emotion Recognition with Self-Distillation," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10094895. keywords: {Training;Emotion recognition;Adaptation models;Transfer learning;Speech recognition;Signal processing;Data models;self-distillation;speech emotion recognition;adaptive inference;efficient deep learning;efficient edge analytics},

- Z. Liu, X. Kang and F. Ren, "Dual-TBNet: Improving the Robustness of Speech Features via Dual-Transformer-BiLSTM for Speech Emotion Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 2193-2203, 2023, doi: 10.1109/TASLP.2023.3282092.

  keywords: {Speech recognition;Acoustics;Feature extraction;Emotion recognition;Speech enhancement;Robustness;Fuses;Speech emotion recognition;affective computing;speech representation learning;feature fusion transformer},

- K. Chauhan, K. K. Sharma and T. Varma, "MNITJ-SEHSD: A Hindi Emotional Speech Database," 2023 International Conference on Communication, Circuits, and Systems (IC3S), BHUBANESWAR, India, 2023, pp. 1-6, doi: 10.1109/IC3S57698.2023.10169497. keywords: {Training;Analytical models;Emotion recognition;Databases;Speech recognition;Feature extraction;Performance analysis;MNITJSEHSD;Speech emotion recognition;Spectral features;Emotional speech database;MFCC;CNN},

- A. Alam, S. Urooj and A. Q. Ansari, "Human Emotion Recognition Models Using Machine Learning Techniques," 2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON), New Delhi, India, 2023, pp. 329-334, doi: 10.1109/REEDCON57544.2023.10151406. keywords: {Emotion recognition;Face recognition;Wearable computers;Electrocardiography;Brain modeling;Feature extraction;Physiology;Human-Computer Interaction;Electrocardiogram (ECG);Electroencephalography (EEG);Galvanic Skin Response (GSR);Emotion Recognition;Machine Learning Techniques},

- W. E. Villegas-Ch, J. García-Ortiz and S. Sánchez-Viteri, "Identification of Emotions From Facial Gestures in a Teaching Environment With the Use of Machine Learning Techniques," in IEEE Access, vol. 11, pp. 38010-38022, 2023, doi: 10.1109/ACCESS.2023.3267007.

  keywords: {Education;Emotion recognition;Face recognition;Real-time systems;Object recognition;Image resolution;Computer vision;Neural networks;Computer vision;emotion recognition;neural networks;teaching},

- M. Norval and Z. Wang, "Creation of an Afrikaans Speech Corpora for Speech Emotion Recognition," 2022 2nd International Conference on Robotics, Automation and Artificial Intelligence (RAAI), Singapore, Singapore, 2022, pp. 193-196, doi: 10.1109/RAAI56146.2022.10092988. keywords: {Human computer interaction;Emotion recognition;Video on demand;Transfer learning;Speech recognition;Companies;Generative adversarial networks;Afrikaans;artificial intelligence;corpora;creative commons;human-computer interaction;South Africa;speech emotion recognition},

- K. Jain, A. Chaturvedi, J. Dua and R. K. Bhukya, "Investigation Using MLP-SVM-PCA Classifiers on Speech Emotion Recognition," 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 2022, pp. 1-6, doi: 10.1109/UPCON56432.2022.9986457. keywords: {Support vector machines;Human computer interaction;Emotion recognition;Speech recognition;Multilayer perceptrons;Feature extraction;Telephone sets;emotion;machine learning;MFCCs;MLPClassifier;SVM;PCA;librosa},

- J. Tharian, R. Nandakrishnan, S. Sajesh, A. V. Arun and C. K. Jayadas, "Automatic Emotion Recognition System using tinyML," 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 2022, pp. 1-4, doi: 10.1109/INCOFT55651.2022.10094330. keywords: {Computer science;Emotion recognition;Neuroscience;Fourier transforms;Psychology;Feature extraction;Mel frequency cepstral coefficient;Emotion Classification;Discrete Fourier Transform (DFT);Deep Learning;Tinyml},

- D. M. Vijayan, A. A. V, G. R, A. N. S. A and R. C. Roy, "Development and Analysis of Convolutional Neural Network based Accurate Speech Emotion Recognition Models," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 2022, pp. 1-6, doi: 10.1109/INDICON56171.2022.10040174. keywords: {Support vector machines;Deep learning;Emotion recognition;Analytical models;Computational modeling;Computer architecture;Feature extraction;Speech Emotion Recognition;CNN;LSTM;Transformer encoder;Accuracy;RAVDESS dataset},

- J. He, P. Liu, K. Zhong, S. Wu, F. Rao and Q. Xu, "Attention-based Mechanism of LSTM network for Speech Emotion Recognition," 2022 International Conference on Machine Learning, Control, and Robotics (MLCR), Suzhou, China, 2022, pp. 101-105, doi: 10.1109/MLCR57210.2022.00027. keywords: {Emotion recognition;Databases;Customer services;Speech recognition;Machine learning;Feature extraction;Telephone sets;SER;Attention mechanism;LSTM;MFCC},

- T. K. Harhare and M. Shah, "Analysis of Acoustic Correlates of Marathi Prosodic Features for Human-Machine Interaction," 2022 International Conference on Engineering and Emerging Technologies (ICEET), Kuala Lumpur, Malaysia, 2022, pp. 1-6, doi: 10.1109/ICEET56468.2022.10007268. keywords: {Human computer interaction;Databases;Machine learning;Linguistics;Acoustics;Software;prosodic features;Acoustic correlates;human-machine interaction;Marathi language},